Sistemi Operativi e Reti

Clusters

Facoltà di Scienze Matematiche Fisiche e Naturali

Corso di Laurea Magistrale in Informatica

Osvaldo Gervasi

ogervasi@computer.org

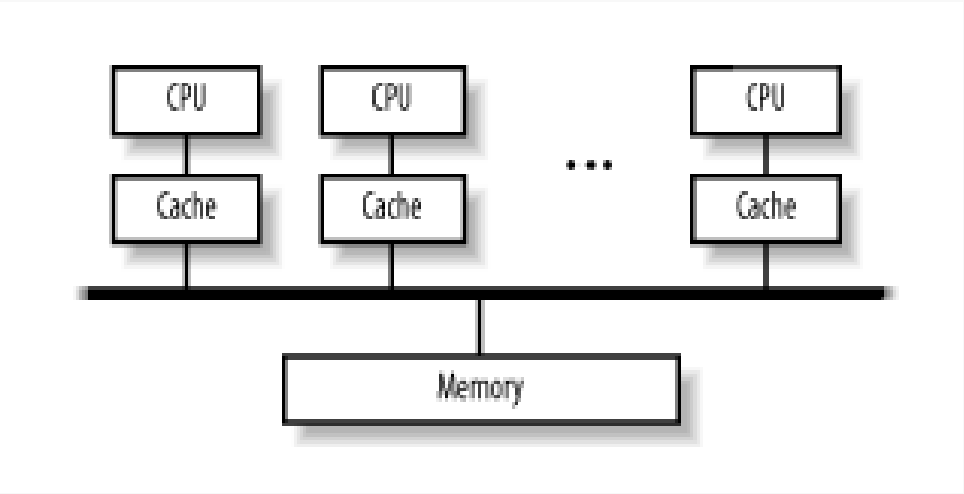
Cluster structure

- It's tempting to think of a cluster as just a bunch of interconnected machines, but when you begin constructing a cluster, you'll need to give some thought to its internal structure.
- This will involve deciding what roles the individual machines will play and what the interconnecting network will look like.

Multiple processors

- In Centralized multiprocessors there are 2 architectural approaches, depending on how the memory is managed:
  - Uniform Memory Access (UMA) architecture
  - Non Uniform Memory Access (NUMA) architecture
- Operating system support is required with either multiprocessor scheme. Fortunately, most modern operating systems, including Linux, provide support for SMP systems, and support is improving for NUMA architectures.

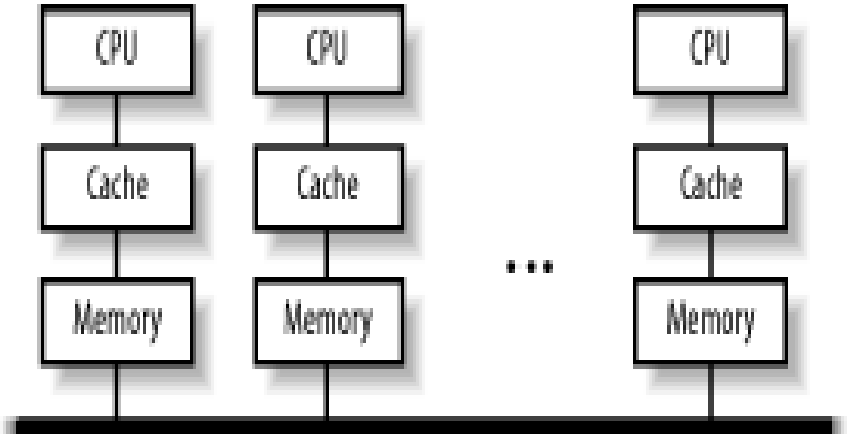
Uniform Memory Access (UMA)



There are various techniques to manage cache consistency. Using snoopy each cache listens to all memory accesses. If a cache contains a memory address that is being written to in main memory, the cache updates its copy of the data to remain consistent with main memory

- UMA machines are also called *symmetric multiprocessors (SMP)*
- There is a common shared memory.
- Identical memory addresses map, regardless of the CPU, to the same location in physical memory.
- Main memory is equally accessible to all CPUs.
- To improve memory performance, each processor has its own cache.
- Two problems: synchronization and cache consistency

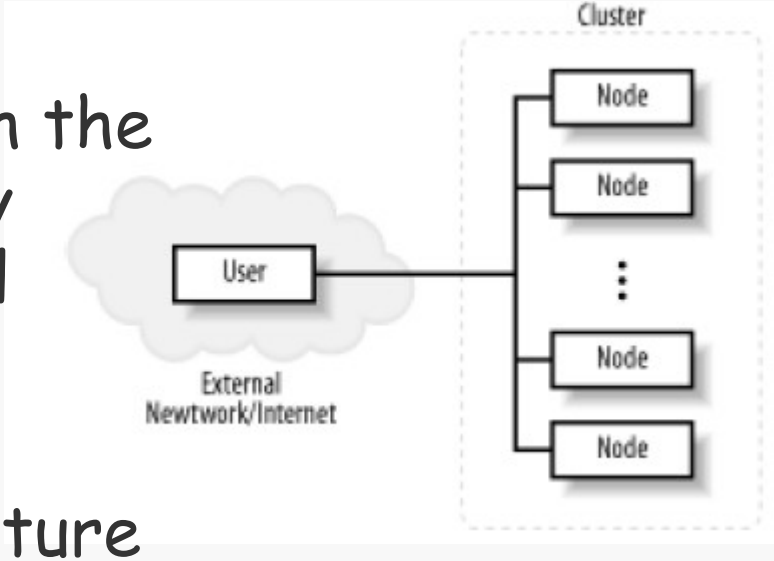
Non Uniform Access Memory (NUMA)



- Each CPU maintains its own piece of memory
- The memory is divided among the processors but each process has access to all the memory
- Each individual memory address, regardless to the processor, still references the same location in memory .
- The memory access is non uniform in the sense that some locations appear to be more slower than other.
- While this arrangement will simplify the synchronization, the problem of cache coherence increases.

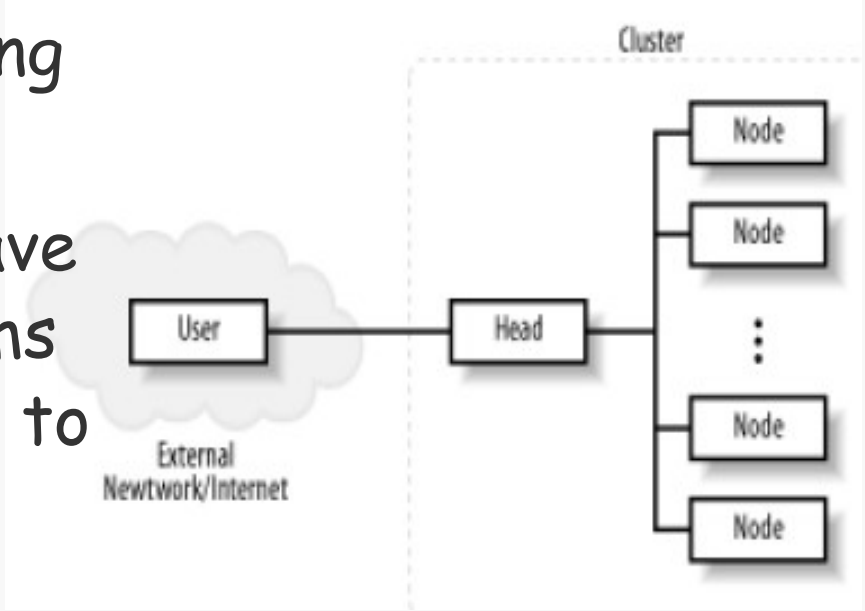
Symmetric Cluster

- Each node can function as an individual computer.
- This is extremely straightforward to set-up
- Just create a sub-network with the individual machines and add any cluster software specific you'll need.
- Each machine will be usable independently (typical architecture of a Network Of Workstation (NOW))
- Cluster management and security can be difficult to implement



Asymmetric Cluster

- One computer is the *head node* or *frontend*. It serves as a gateway between the remaining nodes and the users
- The remaining nodes often have very minimal operating systems and are dedicated exclusively to the cluster
- Since all traffic must pass through the head, asymmetric clusters tend to provide a high level of security.



Asymmetric Clusters

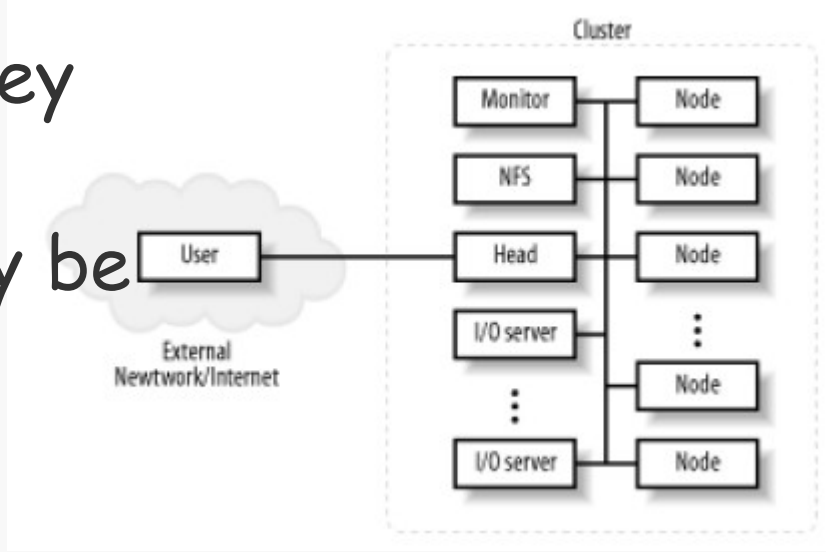
- If the remaining nodes are physically secure and your users are trusted, you'll only need to harden the head node
- The head often acts as a primary server for the remainder of the clusters.
- Since, as a dual-homed machine, it will be configured differently from the remaining nodes, it may be easier to keep all customizations on that single machine. This simplifies the installation of the remaining machines
- The speed of the head may limit the performance of the cluster

Asymmetric Cluster

- Usually the head is a powerful computer respect to the nodes
- Additional server can be incorporated into the Cluster as the number of nodes will increase (i.e: NFS server, DB server, etc)
- I/O represents a particular challenge. It is often desirable to distribute a shared filesystem across a number of machines within the cluster to allow parallel access

Expanded Cluster

- This is a more fully specified cluster
- Network design is another key issue. With small clusters, a simple switched network may be adequate
- With larger clusters, a fully connected network may be prohibitively expensive
- Heterogeneous network are also common (specialized networks)



HA Clusters + LB Clusters

- High Availability (HA) Clusters or failover clusters are often adopted in mission critical applications
- The key issue is *redundancy:* if a primary server goes down a secondary server takes its place
- The Load Balancing Cluster can be implemented using the Round-Robin algorithm of DNS or using some tools that exchange information with the nodes
- It is usual to have both HA and LB in the same Cluster: see the Linux Virtual Server Project (LVSR)

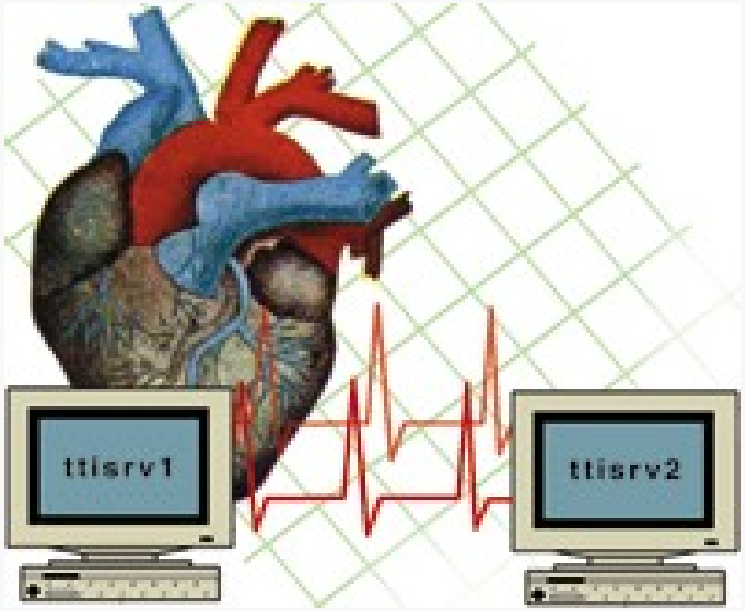
The Linux-HA project

- The basic goal of the High Availability Linux project is to

  Provide a high availability (clustering) solution for Linux which promotes reliability, availability, and serviceability (RAS) through a community development effort.

- The Linux-HA project is a widely used and important component in many interesting High Availability solutions, and ranks as among the best HA software packages for any platform.

Heartbeat



- The Heartbeat program is one of the core components of the Linux-HA (High-Availability Linux) project.
- Heartbeat is highly portable
- Heartbeat is the first piece of software which was written for the Linux-HA project.
- It performs death-of-node detection, communications and cluster management in one process.

Heartbeat

- Heartbeat currently supports a very sophisticated dependency model for n-node clusters. The following types of applications are typical:
  - Database servers
  - ERP applications
  - Web servers
  - LVS director (load balancer) servers
  - Mail servers
  - Firewalls
  - File servers
  - DNS servers
  - DHCP servers
  - Proxy Caching servers

Heartbeat 2

- The Heartbeat ver 2 has been released to overcome several limitations of the ver 1. The main components are:
  - Heartbeat Program
  - LocalResourceManager (is responsible for performing operations on resources, by using ResourceAgent scripts to carry out the work)
  - ClusterResourceManager (CRM)
  - ClusterInformationBase
  - Stonith Daemon (Active fencing mechanism, provides strong data integrity guarantees)

Heartbeat 3

- Since up to release 2.1.4 the messaging layer (Heartbeat proper), the Local Resource Manager, "plumbing" infrastructure and STONITH (now known as **Cluster Glue**), the Resource Agents, and the Cluster Resource Manager (now **Pacemaker**) were all part of a single package named heartbeat, the name was often applied to the Linux-HA project as a whole.
- This generalization is no longer accurate, **the name heartbeat should thus be used for the messaging layer exclusively**.

- Pacemaker is a an Open Source High Availability Resource Manager for both small and large clusters
- In the event of a failure, resource managers like Pacemaker automatically initiate recovery and make sure your application is available from one of the remaining machines in the cluster.
- Cluster's users may never even know there was a problem.

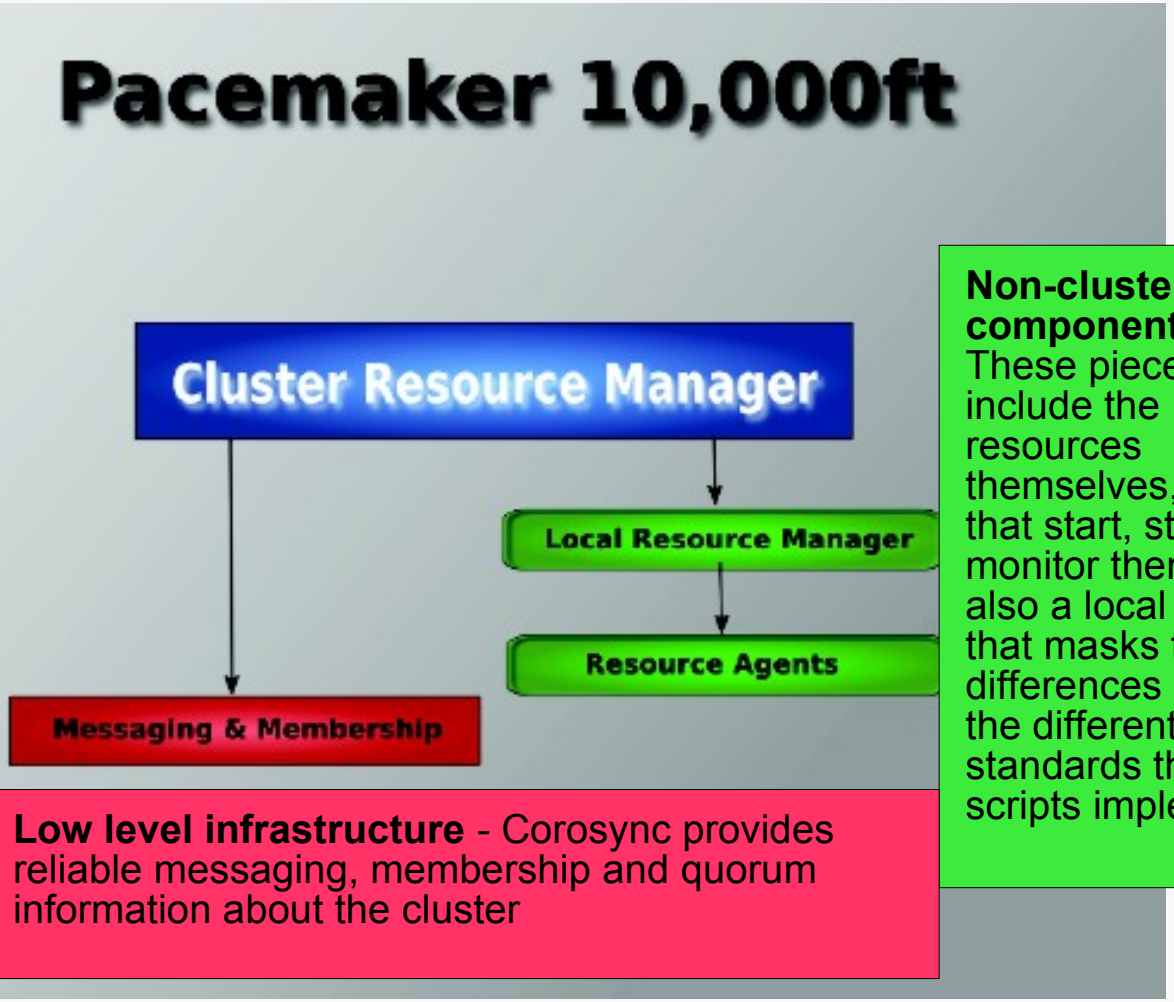
- Pacemaker achieves maximum availability for cluster services by detecting and recovering from node and service-level failures.
- It achieves this by utilizing the messaging and membership capabilities provided by your preferred cluster infrastructure.
- If the startup and shutdown of a service can scripted, Pacemaker can improve its availability. Pacemaker can manage clusters of practically any size and comes with a powerful dependency model for accurately modeling a cluster environment.

HighAvailability

Pacemaker

A scalable High-Availability cluster resource manager

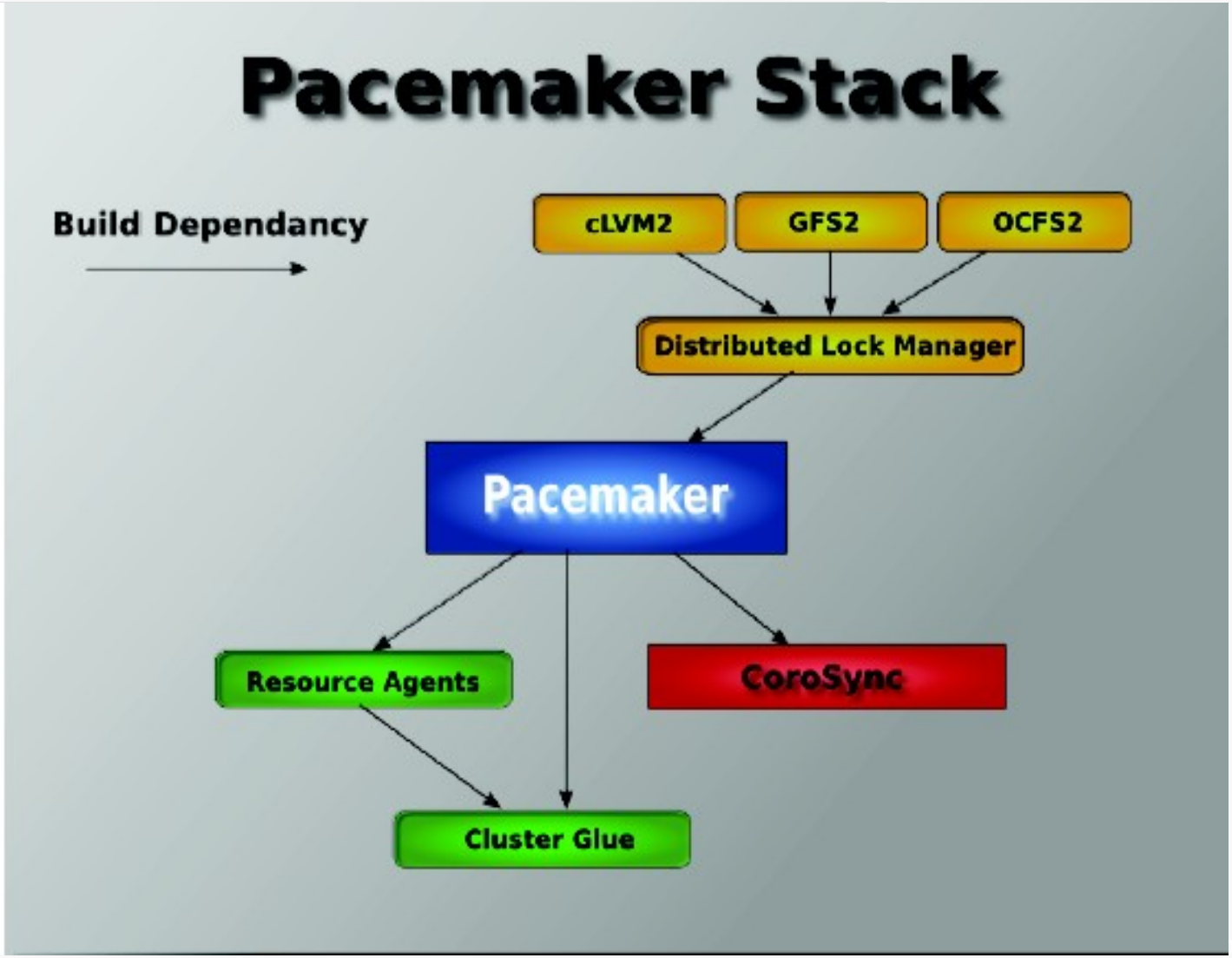
Resource management - Pacemaker provides the brain that processes and reacts to events regarding the cluster. These events include nodes joining or leaving the cluster; resource events caused by failures, maintenance, scheduled activities; and other administrative actions.
Pacemaker will compute the ideal state of the cluster and plot a path to achieve it after any of these events. This may include moving resources, stopping nodes and even forcing them offline with remote power switches.



Low level infrastructure - Corosync provides reliable messaging, membership and quorum information about the cluster

Non-cluster aware components - These pieces include the resources themselves, scripts that start, stop and monitor them, and also a local daemon that masks the differences between the different standards these scripts implement.

Due to recent standardization within the cluster filesystem community, they make use of a common distributed lock manager which makes use of Corosync for its messaging capabilities and Pacemaker for its membership (which nodes are up/down) and fencing services.

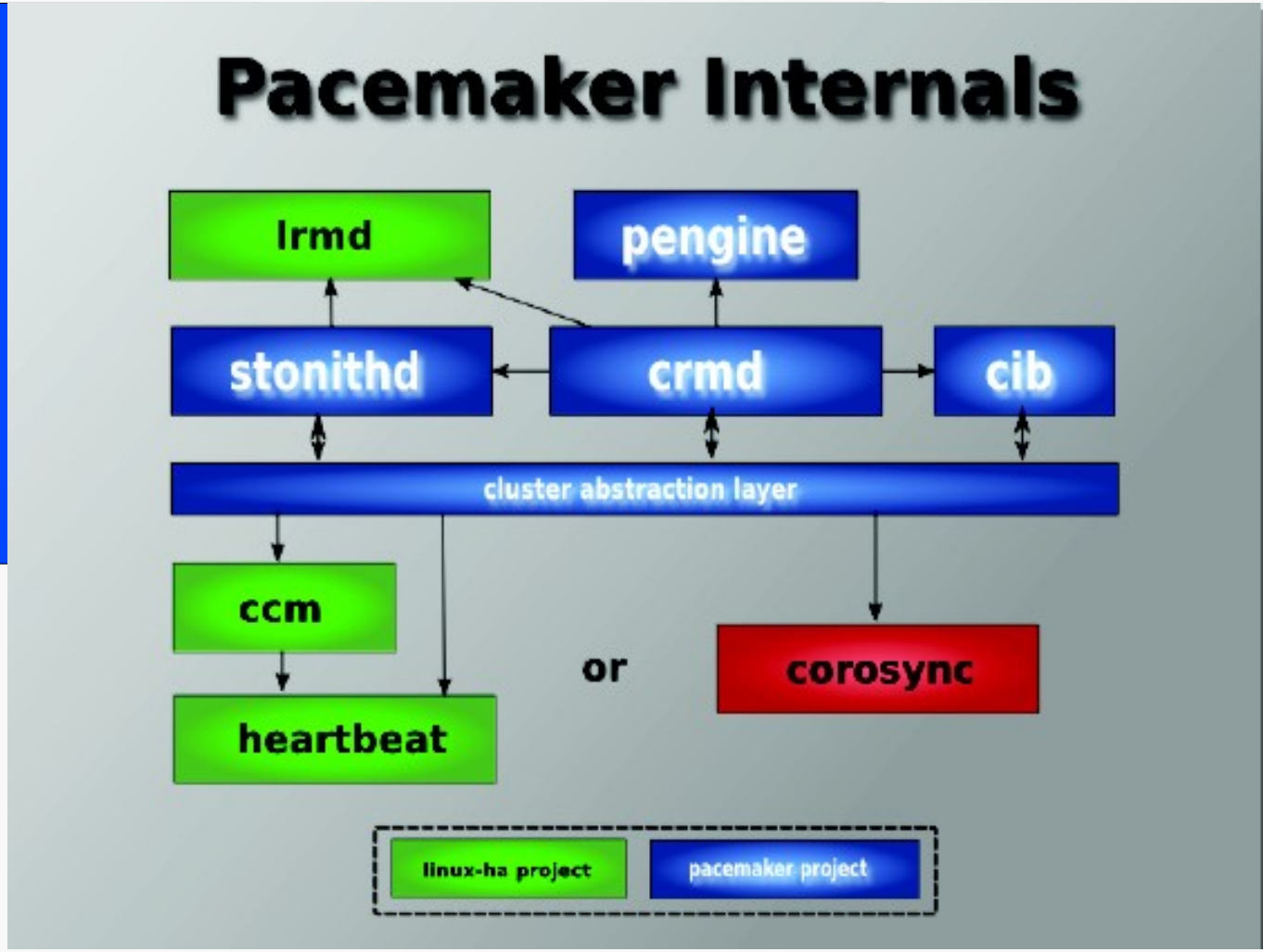


HighAvailability

Pacemaker

A scalable High-Availability cluster resource manager

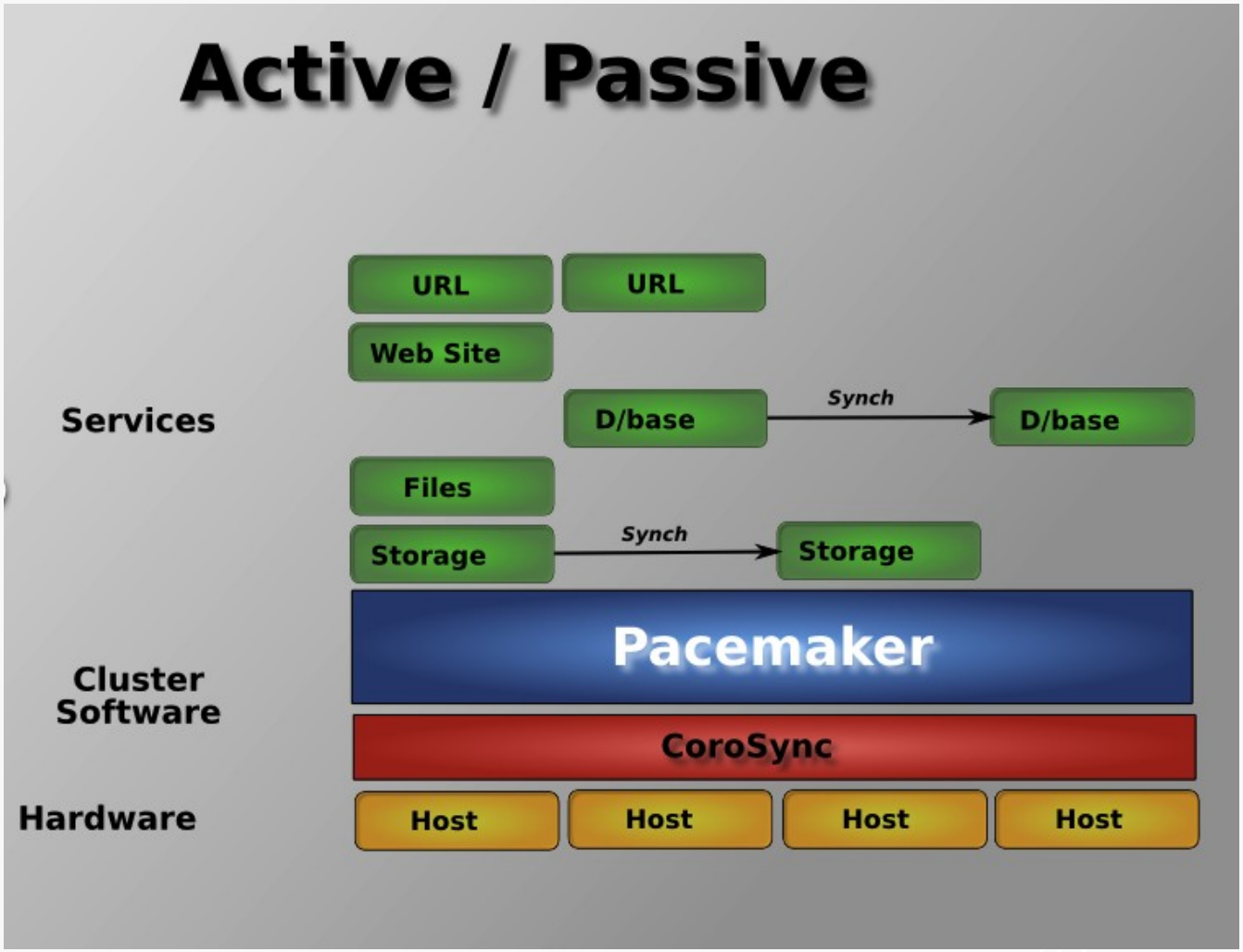
Pacemaker itself is composed of four key components:
• CIB (aka. Cluster Information Base)
• CRMd (aka. Cluster Resource Management daemon)
• PEngine (aka. PE or Policy Engine)
• STONITHd



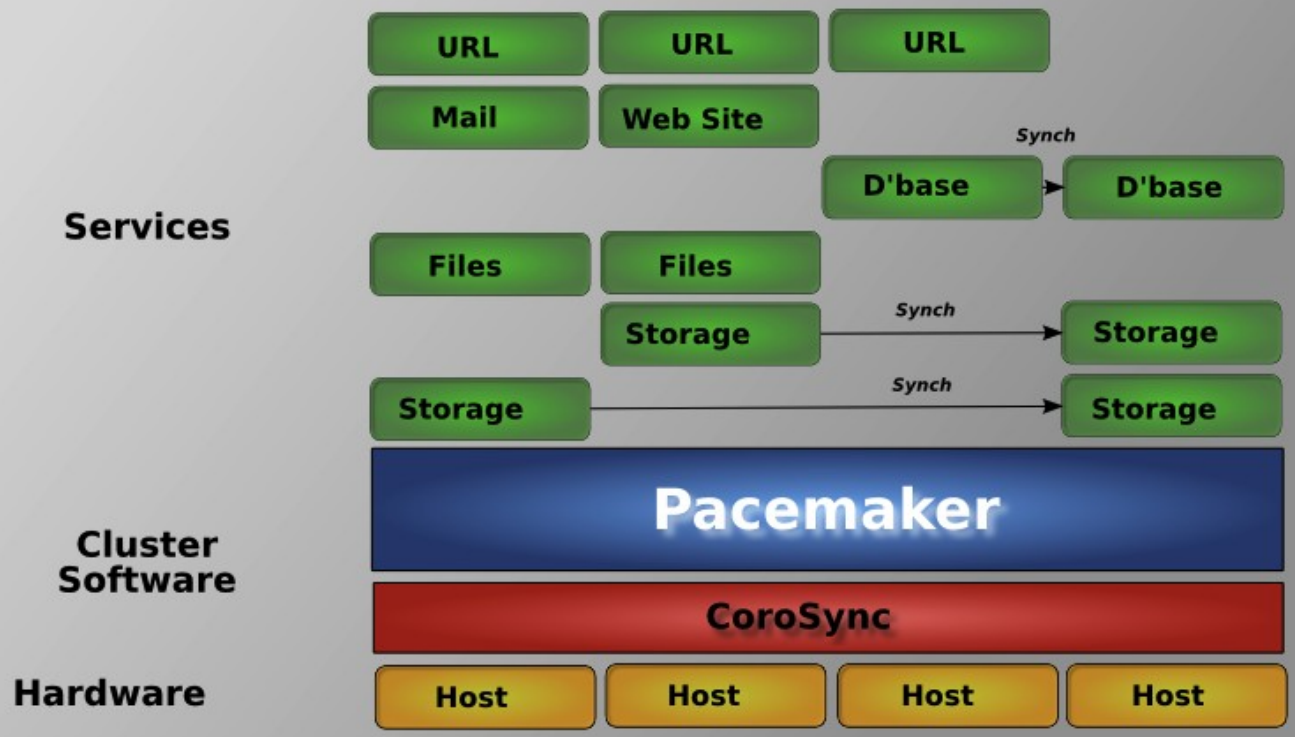
- The CIB uses XML to represent both the cluster's configuration and current state of all resources in the cluster. The contents of the CIB are automatically kept in sync across the entire cluster and are used by the PEngine to compute the ideal state of the cluster and how it should be achieved.
- This list of instructions is then fed to the DC (Designated Co-ordinator). Pacemaker centralizes all cluster decision making by electing one of the CRMd instances to act as a master. Should the elected CRMd process fail (or the node where it runs), a new one is quickly established.
- The DC carries out the PEngine's instructions in the required order by passing them to either the LRMd (Local Resource Management daemon) or CRMd peers on other nodes via the cluster messaging infrastructure (which in turn passes them on to their LRMd process).

- The peer nodes all report the results of their operations back to the DC and based on the expected and actual results, will either execute any actions that needed to wait for the previous one to complete, or abort processing and ask the PEngine to recalculate the ideal cluster state based on the unexpected results.
- In some cases, it may be necessary to power off nodes in order to protect shared data or complete resource recovery. For this Pacemaker comes with STONITHd. STONITH is an acronym for Shoot-The-Other-Node-In-The-Head and is usually implemented with a remote power switch. In Pacemaker,
- STONITH devices are modeled as resources (and configured in the CIB) to enable them to be easily monitored for failure, however STONITHd takes care of understanding the STONITH topology such that its clients simply request a node be fenced and it does the rest.

Active / Passive



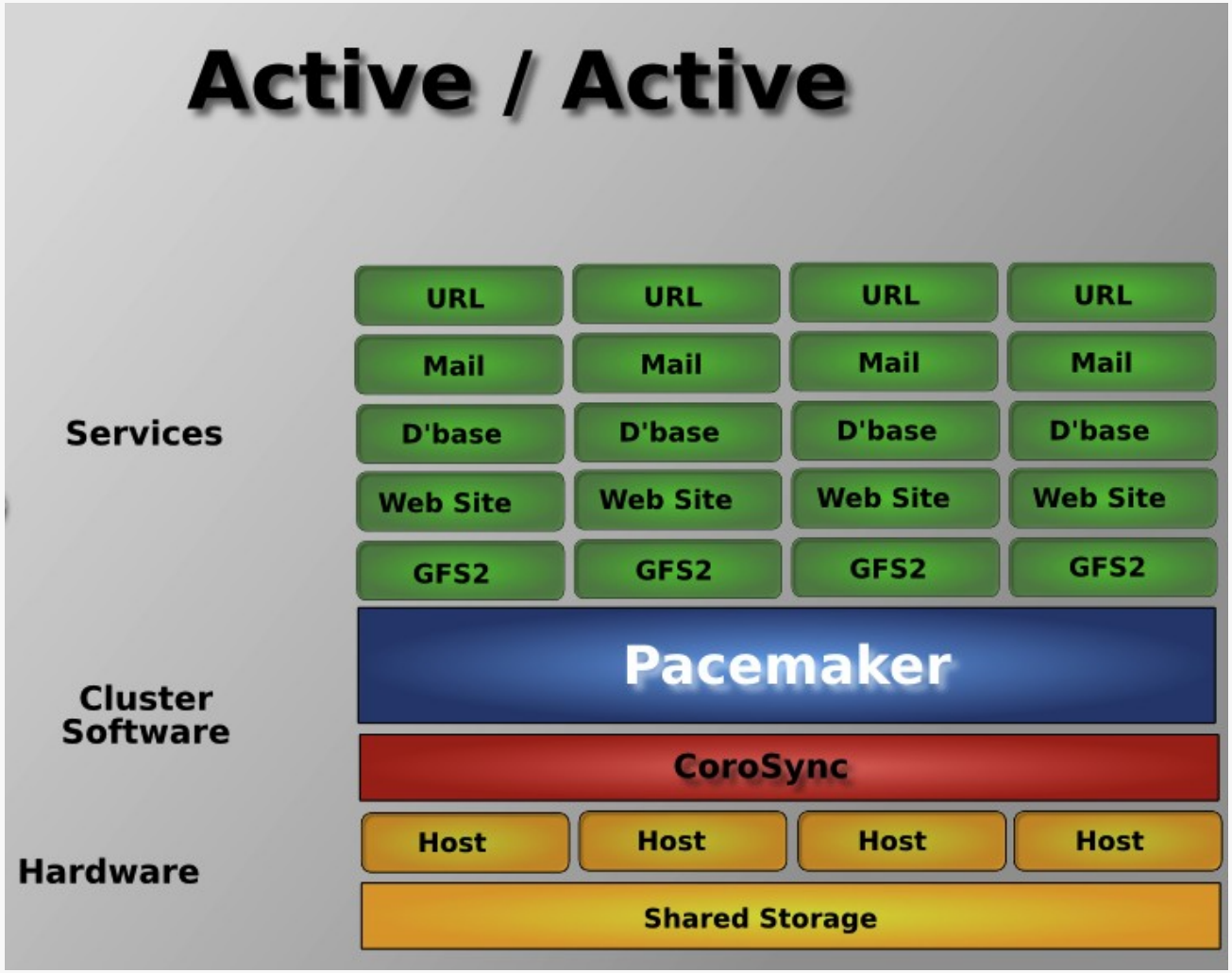
Shared Failover



HighAvailability

Pacemaker

A scalable High-Availability cluster resource manager



Cluster Glue

- **Cluster Glue** is a set of libraries, tools and utilities suitable for the Heartbeat/Pacemaker cluster stack.
- In essence, Glue is everything that is not the cluster messaging layer (Heartbeat), nor the cluster resource manager (Pacemaker), nor a Resource Agent.
- Cluster Glue has been managed as a separate Linux-HA sub-project since its 1.0 release, which coincided with the Heartbeat 2.99 release. Previously, it was a part of the then-monolithic Heartbeat project, and had no separate name.

Cluster Glue components

- Local Resource Manager (LRM): is the interface between the Cluster Resource Manager (Pacemaker) and the resource agents. It is itself not cluster aware, nor does it apply any policies. It simply processes commands received from the Cluster Resource Manager, passes them to resource agents, and reports back success or failure. It particular, the LRM may
  - start a resource;
  - stop a resource;
  - monitor a resource;
  - report a resource's status;
  - list all resource instances it currently controls, and their status.

Cluster Glue components

- **STONITH:** A mechanism for node fencing. In case a node is considered "dead" by the cluster as a whole, STONITH ("Shoot The Other Node In The Head") forcefully removes is from the cluster so it can no longer pose a risk of interacting with other nodes in an uncoordinated fashion.
- **hb_report:** An advanced error reporting utility. hb_report-generated tarballs are frequently requested by the developers to isolate and fix bugs, and are commonly found as attachments to Bugzilla entries.
- **Cluster Plumbing Library:** A low-level library for intra-cluster communications.
- Source code repository: **http://hg.linux-ha.org/glue**

ClusterResourceManager

- PolicyEngine - Computes the NextState of the cluster using information from the ClusterInformationBase
- Transitioner - Attempts to reach the NextState of the cluster by instructing the LocalResourceManager on various remote nodes to perform actions on its resources.
- LocalResourceManager - In charge of actually performing the start/stop actions

- The ClusterResourceManager uses IPC to send messages to its subsystems and HeartbeatMessages for communication with the ClusterResourceManagerDaemon or DesignatedCoordinator on other ClusterNodes

OpenAIS

- It is possible to write redundant applications that tolerate hardware, operating system, and application faults.
- Cluster software developers can write plugins to use the infrastructure provided by OpenAIS.
- OpenAIS implements the communication system between the nodes, so that the CRM can interact with them.

OpenAIS

- OpenAIS is an open source implementation of the SA Forum (www.saforum.org) Application Interface Specification.
- The Application Interface Specification is a software API and policies which are used to develop applications that maintain service during faults.
- The API consists of Availability Management Framework (AMF) which provides application failover, Cluster Membership (CLM), Checkpointing (CKPT), Eventing (EVT), Messaging (MSG), and Distributed Locking (DLOCK).

OpenAIS

- The project currently implements APIs to improve availability by reducing MTTR.
- APIs available are cluster membership, application failover, checkpointing, eventing, distributed locking, messaging, closed process groups, and extended virtual synchrony passthrough.
- The major focus of high availability in the past has been to mask hardware faults. Faults in other components of the system have gone unsolved until AIS.

OpenAIS

- AIS can mask many types of faults in applications, middleware, operating systems, or even hardware by providing a simple framework for allowing developers to create redundant applications.
- These redundant applications can be distributed over multiple nodes such that if any one node faults, another node can recover.
- Application programmers develop applications to periodically record their state using the checkpointing service.

OpenAIS

- When an active application fails, a standby application recovers the state of the application. This technique, called **stateful application failover**, provides the fundamental difference between openais and other systems that have come before it.
- With stateful application failover, the end-application user doesn't have to reload the application or redial a telephone. The full state is recorded, so the end-application user sees no interruption in service.

OpenAIS

- Because programmers can now distribute applications across multiple processes or nodes, a mechanism must exist for them to communicate.
- This mechanism is provided by two services.
- The **event service** provides a publish/subscribe model for events.
- The **messaging service** provides end to end messaging.
- Finally a mechanism to synchronize access is provided by the **distributed lock service**.

OpenAIS

- The openais project also provides a group messaging toolkit called EVS.
- The EVS service implements a messaging model known as Extended Virtual Synchrony.
- This model allows one sender to transmit to many receivers.
- Certain guarantees are provided for message and membership delivery which make virtual synchrony ideal for developing distributed applications.

Corosync
The Corosync Cluster Engine

Corosync

- Corosync Cluster Engine has been implemented as an evolution of OpenAIS, to solve the problems observed working with OpenAIS, PeaceMaker and Apache Qpid
- Corosync approaches High Availability by ensuring every redundant server in the system maintains a redundant copy of information used to make decisions for the application.
- This approach, called a distributed state machine, is simple to use.

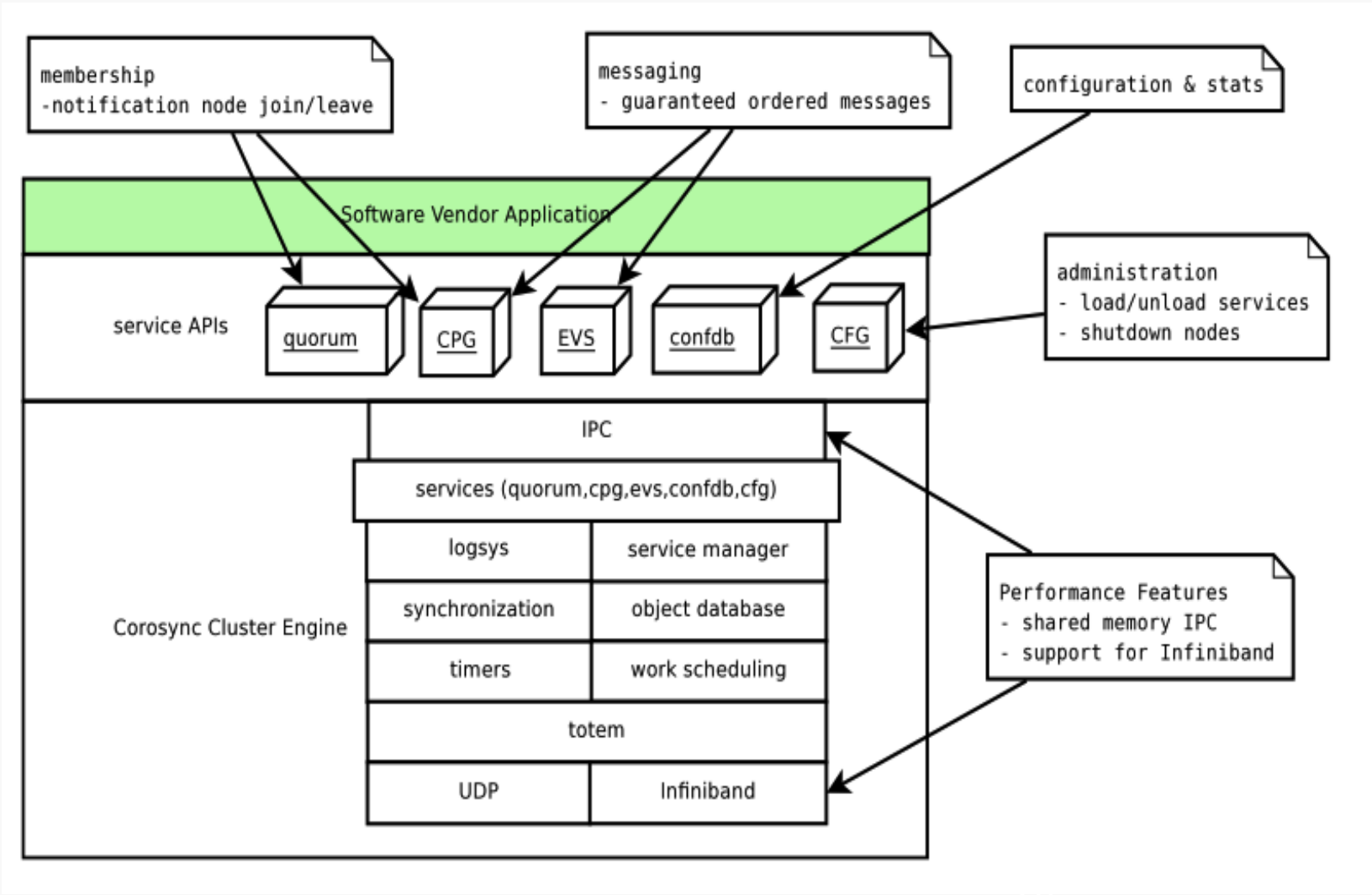
Corosync
The Corosync Cluster Engine

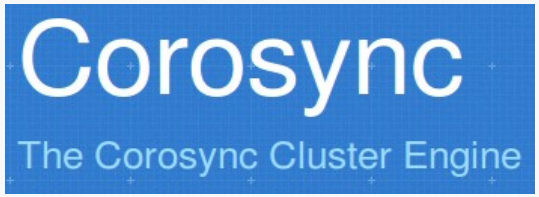
Corosync

- In a typical state machine, software designers call functions which change the state of the application.
- Using Corosync, software designers send messages instead of call functions.
- When these messages are delivered, the state machine on all nodes changes its state in an ordered and consistent fashion.
- Corosync is highly tuned and designed for performance. Special consideration has been taken to minimize memory end context switching

Corosync
The Corosync Cluster Engine

Corosync





Corosync

- Closed Process Group provide a membership within applications. When an application joins a process group, all applications in that process group are sent a membership change with the Process ID and the Node ID of the application. Such membership information can be used for making application decisions.
- Once an application is joined to a process group, it may send and receive messages.
- A sent message is delivered to all members of the process group, which then change their distributed state machine

LVSR

- The Linux Virtual Server is a highly scalable and highly available server built on a cluster of real servers, with the load balancer running on the Linux operating system.
- The architecture of the server cluster is fully transparent to end users, and the users interact as if it were a single high-performance virtual server
- The Linux Virtual Server as an advanced load balancing solution can be used to build highly scalable and highly available network services, such as scalable web, cache, mail, ftp, media and VoIP services.

Cluster management

- Cluster management is to monitor and administrate all the computers in a computer cluster.
- It covers a wide range of functionality, such as resource monitoring, cluster membership management, reliable group communication, and full-featured administration interfaces.
- One of the advantages of a cluster system is that it has hardware and software redundancy, because the cluster system consists of a number of independent nodes, and each node runs a copy of operating system and application software.

Cluster management

- Cluster Management can help achieve high availability by detecting node or daemon failures and reconfiguring the system appropriately, so that the workload can be taken over by the remaining nodes in the cluster.
- LVS management is composed by:
  - Cluster monitoring
  - Administration interface – Provides the following functions:
    - add new servers to increase the system throughput or remove servers for system maintenance, without bringing down the whole system service
    - monitor the traffic of LVS cluster and provide statistics

Cluster monitoring

- The major work of cluster monitoring in LVS is to monitor the availability of real servers and load balancers, and reconfigure the system if any partial failure happens, so that the whole cluster system can still serve requests
- To monitor the availability of real servers, there are two approaches, one is to run service monitoring daemons at the load balancer to check server health periodically, the other is to run monitoring agents at real servers to collect information and report to the load balancer.

daemons or servers automatically.

Cluster monitoring

- The service monitor usually sends service requests and/or ICMP ECHO_REQUEST to real servers periodically, and remove/disable a real server in server list at the load balancer if there is no response in a specified time or error response, thus no new requests will be sent to this dead server.
- When the service monitor detects the dead server has recovered to work, the service monitor will add the server back to the available server list at the load balancer.
- Therefore, the load balancer can mask the failure of service daemons or servers automatically.

Cluster monitoring

- In the monitoring agent approach, there is also a monitoring master running at the load balancer to receive information from the agents.
- The monitoring master will add/remove servers at the load balancer based on the availability of agents, can also adjust server weight based on server load information

Load balancer

- The load balancer is the core of a server cluster system, and it cannot be a single failure point of the whole system.
- In order to prevent the whole system from being out of service because of the load balancer failure, we need setup a backup (or several backups) of the load balancer, which are connected by heartbeat or Virtual Router Redundancy Protocol (VRRP).
- Two heartbeat daemons run on the primary and the backup respectively, they heartbeat the message like "I'm alive" each other through serial lines and/or network interfaces periodically.

VRRP: Virtual Router Redundancy Protocol

- The Virtual Router Redundancy Protocol (VRRP) is a non-proprietary redundancy protocol described by the RFC 3768, designed to increase the availability of the default gateway servicing hosts in the same subnet.
- This increased reliability is achieved by advertising a "virtual router" (an abstract representation of master and backup routers acting as a group) as a default gateway to the host(s) instead of one physical router.
- Two or more physical routers are then configured to stand for the virtual router, with only one doing the actual routing at any given time.
- If the current physical router that is routing the data on behalf of the virtual router fails, an arrangement is made for another physical router to automatically replace it.

Load balancer

- When the heartbeat daemon of the backup cannot hear the heartbeat message from the primary in the specified time, it will take over the virtual IP address to provide the load-balancing service.
- When the failed load balancer comes back to work, there are two solutions, one is that it becomes the backup load balancer automatically, the other is the active load balancer releases the VIP address, and the recover one takes over the VIP address and becomes the primary load balancer again.

Load balancer

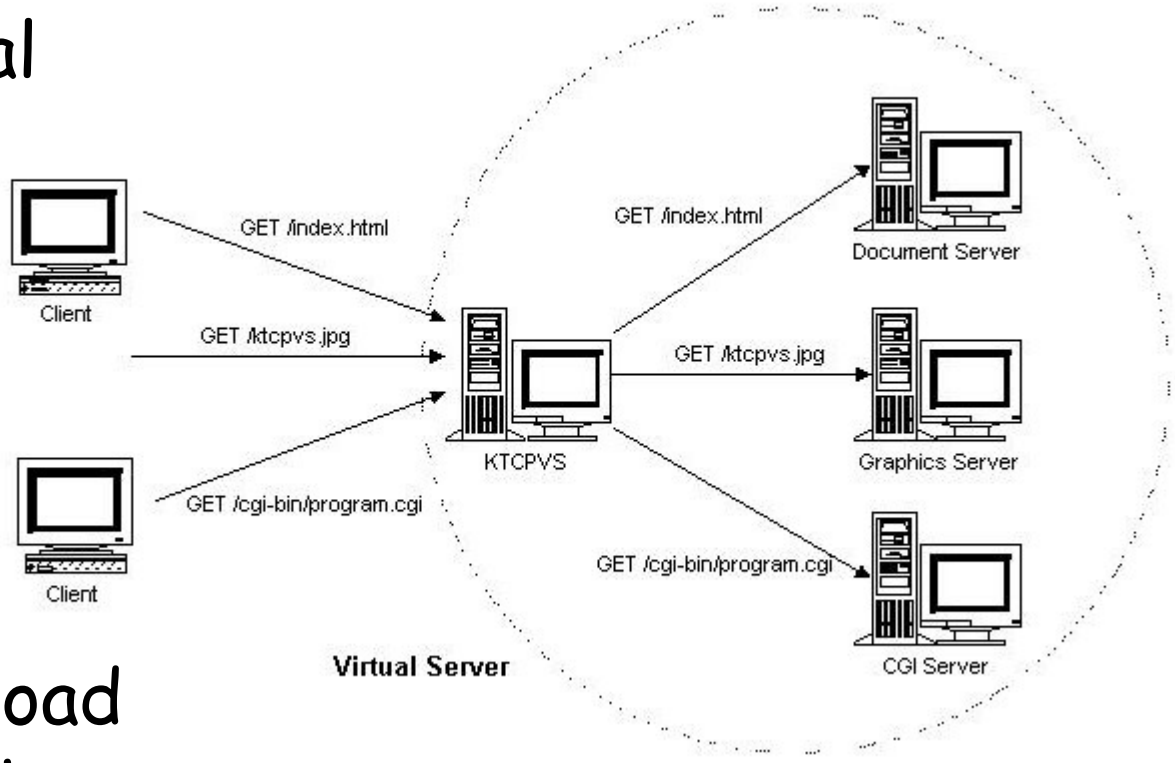
- The primary load balancer has state of connections, i.e. which server the connection is forwarded to. If the backup load balancer takes over without those connections information, the clients have to send their requests again to access service.
- In order to make load balancer failover transparent to client applications, has been implemented connection synchronization in IPVS, the primary IPVS load balancer synchronizes connection information to the backup load balancers through UDP multicast.
- When the backup load balancer takes over after the primary one fails, the backup load balancer will have the state of most connections, so that almost all connections can continue to access the service through the backup load balancer.

IPVS

- LVS uses IPVS for a transport layer load balancing inside the Linux kernel (Layer 4 switching).
- IPVS is running on a host (load balancer) that will direct the UDP/TCP based services to the real servers
- The services of the real servers appear as a virtual service on a single IP address.

KTCPVS

- Kernel TCP Virtual Server



- It implements application-level load balancing inside the Linux kernel, so called Layer-7 switching

Cluster kits

- The installation of a cluster is a complicated process that can be facilitated by *cluster kits*, software packages that automate the installation process
- A cluster kit provides all the software you are likely to need in a single distribution
- Cluster kits tend to be very complete. For example, the OSCAR distribution contains both PVM and two versions of MPI. If some software isn't included, you can probably get by without it.

Cluster kits

- Some kits have a Linux distribution included in the package (e.g., Rocks), while others are installed on top of an existing Linux installation (e.g., OSCAR).
- Even if Linux must be installed first, most of the configuration and the installation of needed packages will be done by the kit.
- There are two problems with using cluster kits:
  - First, cluster kits do so much for you that you can lose touch with your cluster
  - In making everything work together, kit builders occasionally have to do things a little differently

Cluster kits

- Consider that while a cluster kit can get you up and running quickly, you will still need to learn the details of the individual software.
- You should follow up the installation with a thorough study of how the individual pieces in the kit work.
- For most beginners, the single advantage of being able to get a cluster up and running quickly probably outweighs all of the disadvantages.

Cluster kits

- While other cluster kits are available, the three most common kits for Linux clusters are
  - NPACI Rocks (CentOS)
  - OSCAR (Fedora 7/8, RedHatEnterpriseLinux (RHEL), OpenSuse, Debian)
  - Scyld Beowulf
- While Scyld Beowulf is a commercial product available from Penguin Computing, an earlier, unsupported version is available for a very nominal cost from http://www.linuxcentral.com/
- However OSCAR and Rocks are the best products

NPACI Rocks

- NPACI (National Partnership for Advanced Computational Infrastructure) Rocks is a collection of open source software for creating a cluster built on top of Red Hat Linux.
- Rocks takes a cookie-cutter approach. To install Rocks, begin by downloading a set of ISO images from http://rocks.npaci.edu/Rocks/ and use them to create installation CD-ROMs.
- Next, boot to the first CD-ROM and answer a few questions as the cluster is built. Both Linux and the clustering software are installed.

NPACI Rocks

- The installation should go very quickly. In fact, part of the Rocks' management strategy is that, if you have problems with a node, the best solution is to reinstall the node rather than try to diagnose and fix the problem.
- Depending on hardware, it may be possible to reinstall a node in under 10 minutes. When a Rocks installation goes as expected, you can be up and running in a very short amount of time.
- However, because the installation of the cluster software is tied to the installation of the operating system, if the installation fails, you can be left staring at a dead system and little

OSCAR

- OSCAR, from the Open Cluster Group, uses a different installation strategy. With OSCAR, you first install Linux (but only on the head node) and then install OSCAR—the installations of the two are separate
- This makes the installation more involved, but it gives you more control over the configuration of your system, and it is somewhat easier to recover when you encounter installation problems
- And because the OSCAR installation is separate from the Linux installation, you are not tied to a single Linux distribution.

Rocks vs OSCAR

- Rocks uses a variant of Red Hat's Anaconda and Kickstart programs to install the compute nodes Thus, Rocks is able to probe the system to see what hardware is present
- To be included in Rocks, software must be available as an RPM and configuration must be entirely automatic
- As a result, with Rocks it is very straightforward to set up a cluster using heterogeneous hardware

Rocks vs OSCAR

- OSCAR, in contrast, uses a system image cloning strategy to distribute the disk image to the compute nodes.
- With OSCAR it is best to use the same hardware throughout your cluster.
- Rocks requires systems with hard disks
- OSCAR's thin client model is designed for diskless systems.

Rocks vs OSCAR

- Rocks and OSCAR include a variety of software and build complete clusters.
- In fact, most of the core software is the same for both OSCAR and Rocks.
- However, there are a few packages that are available for one but not the other. For example, Condor is readily available for Rocks while LAM/MPI is included in OSCAR.
- Clearly, Rocks and OSCAR take orthogonal approaches to building clusters.

Rocks vs OSCAR

- OSCAR scales well over Linux distributions, Rocks scales well with heterogeneous hardware
- No one approach is better in every situation
- A novice can probably build a Rocks cluster a little faster than an OSCAR cluster. But if you want greater control over how your cluster is configured, you may be happier with OSCAR in the long run.
- Typically, OSCAR provides better documentation, although Rocks documentation has been improving

CD-ROM-based Clusters

- If you just want to learn about clusters, only need a cluster occasionally, or can't permanently install a cluster, you might consider one of the CD-ROM-based clusters
- With these, you create a set of bootable CD-ROMs, sometimes called "live filesystem" CDs.
- When you need the cluster, you reboot your available systems using the CD-ROMs, do a few configuration tasks, and start using your cluster
- The cluster software is all available from the CD-ROM and the computers' hard disks are unchanged When you are done, you simply remove the CD-ROM and reboot the system to return to the operating system installed on the hard disk

CD-ROM-based Clusters

- Clearly, this is not an approach to use for a high-availability or mission-critical cluster, but it is a way to get started and learn about clusters.
- It is a viable way to create a cluster for shortterm use. For example, if a computer lab is otherwise idle over the weekend, you could do some serious calculations using this approach.
- There are some significant difficulties with this approach, most notably problems with storage.
- It is possible to work around this problem by using a hybrid approach—setting up a dedicated system for storage and using the CD-ROM-based systems as compute-only nodes.

CD-ROM-based Clusters

- Several CD-ROM-based systems are available:
  - ClusterKnoppix
    http://bofh.be/clusterknoppix/
  - Bootable Cluster CD (BCCD)
    http://bccd.cs.uni.edu/

BCCD

- BCCD was developed by Paul Gray as an educational tool
- If you want to play around with a small cluster, BCCD is a very straightforward way to get started.
- On an occasional basis, it is a viable alternative
- You need to download the ISO CD images from the web site and burn a CD for each machine.
- Next, boot each machine in your cluster from the CD-ROM

BCCD

- You'll need to answer a few questions as the system boots.
  - First, you'll enter a password for the default user, *bccd*
  - Next, you'll answer some questions about your network. The system should autodetect your network card. Then it will prompt you for the appropriate driver. If you know the driver, select it from the list BCCD displays. Otherwise, select "auto" from the menu to have the system load drivers until a match is found. If you have a DHCP and DNS server available on your network, this will go much faster

BCCD

- Once the system boots, log in to complete the configuration process.
- When prompted, start the BCCD heartbeat process.
- Next, run the utilities
  - *bccd-allowall* and
  - *bccd-snarfhosts*.
- The first of these collects hosts' keys used by SSH and the second creates the *machines* file used by MPI. You are now ready to use the system.

Cluster Hardware

Design decisions

Hardware selection

- The final destination of the cluster dictates the cluster architecture and the hardware characteristics of the machines
- Nonetheless the badgetary constraints may force to less ideal solutions to adopt
- If possible select identical systems for the computer nodes -life will be much simpler!
  - Compatibility tests will be performed only on a single machine, the other will be cloned
  - Resource balancing easier and more effective
  - Maintenance and repair is simpler

Building a cluster

- The following strategies can be adopted:
  - scrounge for existing computers
    - cheapest solution
    - hardware and software problems
  - buy new pre-assembled computers
    - simplest approach if money isn't a concern
    - best approach for mission-critical environments (money not an issue, short time available)
  - buy the parts and assemble your own
    - cheaper solution
    - high performance and reliability
    - allows customization

Node hardware

- The configuration must be limited to the essential, particularly for dedicated clusters
- CPU and motherboard
  - Represent the crucial components of the environment
  - For high performances (critical factors: processor clock rate, cache size, bus speed, memory capacity, disk access speed, network latency) the two parts need to be totally compatible.
  - Clock rate should be compared considering the total cost of the nodes
  - The **latest model** usually **isn't the right choice**

Node hardware

- Memory and cache
  - The more memory and cache in your system, the better
  - The faster the processor, the more RAM is needed (crude rule: one byte per FLOPS is needed)
  - Paging creates a severe performance penalty and should be avoided
  - Diskless cluster are an important solution in some circumstances (reduced costs, easy maintenance).
    - The averave life of a disk is reported in 300.000 hours (34 years). In a cluster of 100 machines you replace 3 disks per year. In a cluster of 12000 nodes you have a failure every 25 hours!

Node hardware

- The downside to consider is that the configuration is more complex and the network load increases significantly (paging over the network will be devastating to performance!)
- Disk-based systems are more versatile and more forgiving.

- If you are buying a disk, there are three issues: interface type (EIDE, SATA, SCSI), disk latency (a function of rotational speed) and the capacity. The Serial ATA offers an interesting price/performan-ce ratio, EIDE is the cheapest solution. Almost all current drives rotate at 7200 rpm, while a few rotate at 10000 rpm

Node hardware

- A CD/DVD reader (and a floppy) could be useful to be added, considered the low cost and their usefulness.
- The CD-ROM or the floppy are used to customize and initiate the network installs
- These devices are useful to recover some file system or disk failures
- The only compelling reason not to include CD-ROM or floppy is a lack of space in a truly minimal system

Node hardware

- Monitors Keyboard and mouse
  - Many minimal systems elect not top include monitors, keyboards and mouses, but rely on the network to provide the local connectivity
  - Several problems can be encountered with these *hardless* systems: depending on the system BIOS you may not be able to boot a system without a display or a keyboard attached. Often there are CMOS options that allow to override the tests.
  - For small clusters a Keyboard video mouse (KVM) switch is the solution. The switch allow you to determine which machine is connected. You'll be able to connect only to one of the machines at the time, but you can cycle among them clicking a button

Node hardware

- There are different KVM switches available
- The cables are often not included with the switch and are overpriced
- The KVM switch assumes that individual machines have a monitor adapter
- The alternative is to use serial consoles
- With a fair amount of work every Linux system can be reconfigured to work this manner
- Additional hardware is available that will allow you to multiplex serial connections from a number of machines (see the Remote Serial Console Howto).

Node hardware

- Adapters, power supplies and cases
  - A Video adapter is necessary. The Network adapters is also a key component and must be compatible with the cluster network
  - The power supply must be choosen carefully in order to optimize the cluster operations and the maintenance costs.
  - Google is using less powerful machine in order to balance computational needs with the total operational costs (considering also the cooling needs)

Cluster head and servers

- The head and the additional servers should be complete systems, since il will add little to the overall costs, but will facilitate customizing and maintaining these systems
- The head node must be dual-homed (unless, as suggested for security reasons, a separate host acting as a firewall is used)
- Particularly useful will be a disk server designed to provide a reliable and fast access to large amount of storage

Cluster network

- For commodity clusters, networking is often the weak link
- The two key factor are bandwidth and latency
  - Applications that move large amount of data need bandwidth
  - Real time applications and applications that have lots of interactions among nodes, minimizing latency is critical
- High-end Ethernet is the common choice for clusters
- For some critical low-latency applications, you have to consider special low-latency hardware.

Cluster network

- Myrinet is the common alternative to Ethernet, from Myricom Inc.
- Myrinet is a proprietary solution providing high-speed bidirectional connectivity (2+2Gbps or 10+10 Gbps) and low latency (2.6-2.0 micro-seconds)
- Myrinet 2000 is an alternative to Gigabit Ethernet Clusters, while Myri-10G is an alternative to 10GEthernet
- The same cables of the corresponding Ethernet are used

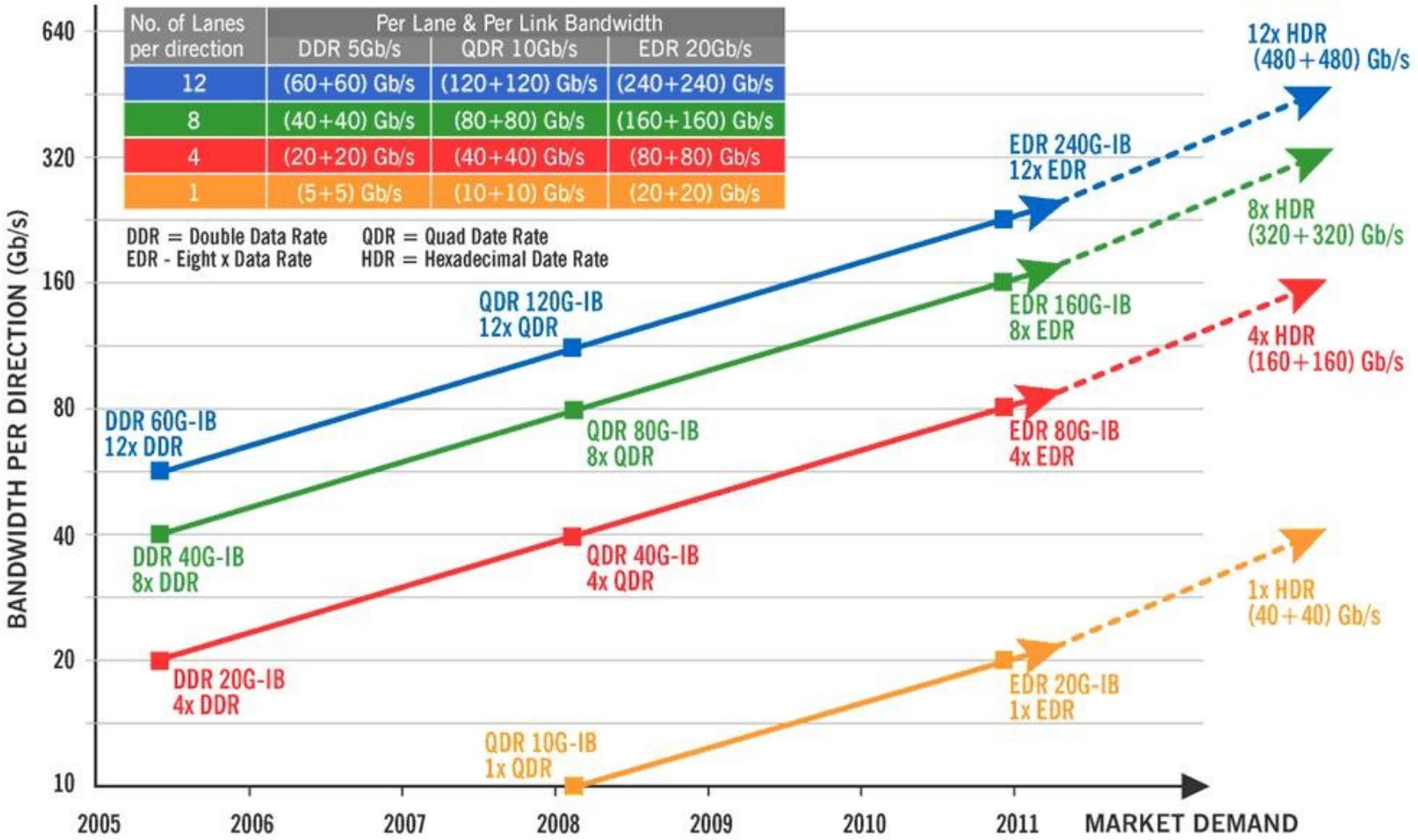
Cluster Network

Product series	Myrinet-2000	Myri-10G
Full-duplex data rate for the links, NIC ports, and switch ports	2+2 Gigabits/s	10+10 Gigabits/s
Link cables	LC-connectorized duplex multimode fiber to 200m	Selected 10-Gigabit Ethernet cables, copper and fiber
NICs	Single-port and dual-port PCI-X	Single-port PCI-Express, dual-protocol 10G Myrinet or 10G Ethernet
Switches	Based on 16-port and 32-port crossbar switches	Based on 16-port crossbar switches
Switch networks	Up to 256 host ports with a single "Network in a Box" component, and up to tens of thousands of hosts by combining these components	Up to 128 host ports with a single "Network in a Box" component, and up to tens of thousands of hosts by combining these components
Interoperability	Gigabit Ethernet	10-Gigabit Ethernet
Myrinet software support	Myrinet Express (MX-2G) or GM-2	Myrinet Express (MX-10G)
MX or MPI latency	2.6µs–3.2µs	2µs
MX unidirectional data rate	247 MBytes/s (one-port NICs) 495 MBytes/s (two-port NICs)	1.2 GBytes/s
TCP/IP (MX ethernet emulation) data rate	1.98 Gbits/s (one-port NICs) 3.95 Gbits/s (two-port NICs)	9.6 Gbits/s

Cluster networks

- Competitive alternatives are
  - CLAN from Emulex
  - QsNet from Quadrix
  - Infiniband from the Infiniband Consortium
- The problem of these alternatives is the high cost. You can triple the cost of a node
- Gigabit Ethernet is better served using an embedded adapter rather than an additional PCI board, since Gigabit can swamp the PCI bus. Conversely for FastEThernet a separate adapter is preferred since the embedded adapter may steal clock cycles from your application.

InfiniBand Link Speed Roadmap



Cluster networks

- Very high-performance clusters may have two parallel networks. One is used for messages passing among the nodes, while the second is used for the network file system
- In the past, elaborate technologies, architecture and topologies have been developed to optimize communications
- Channel bonding uses multiple interfaces to multiplex channels for higher bandwidth.
- Hypercube topologies have been used to minimize communication path length

Environment

- An accurate planning of the physical space, wiring, cooling and physical access is required
- The distribution among power circuits is crucial
- Ventilation must be preserved
- Cable manage is also a concern
- Ideally power and data cables must be separated
- Standard equipment racks are very nice. Cabling is greatly simplified, but things are closely packed and heat could be a problem

Power and air conditioning

- The power required must be evaluated carefully (considering a +50% for safety)
- The quality of the power is an issue
- A UPS can be considered both for providing a fault tolerance for short power interruptions and for stabilize the power
- These systems can be managed through serial lines and SNMP

Linux for clusters

Installing Linux

- You need to install a Linux version tested for the environment you want to implement
- Common Linux distributions:
  - Scientific Linux
  - CERN Scientific Linux
  - Fedora
  - RedHat Enterprise
  - Debian
  - Ubuntu
  - Mandriva
  - CentOS
  - SUSE

Head node

- The distribution is not important for the user because the head node will be used only for preparing and submitting job.
- The main issue is the software compatibility with the middleware used. The distribution must be maintained by developers (bug fixes)
- It will be prefereable to start with a clean installation of the OS and install only the packages used
- Compilers, libraries and editors are usually mandatory.

Configuring services

- Once the basic installation is completed, you'll need to configure the system.
- Many of the tasks are the same of the other systems
- OSCAR or Rocks perform these operation for you
- DHCP:
  - Dynamic Host Configuration Protocol is used to supply network configuration parameters, including IP addresses and host names, to the clients, as they boot.

DHCP

- With clusters the head node is usually configured as DHCP server, while the compute nodes as DHCP clients.
  - It simplify the installation of the compute nodes since the information DHCP provides are those that are different from node to node
  - It is much easier to change the configuration of the network
  - The server configuration file, /etc/dhcpd.conf controls the information distributed to the clients
  - You need to include a subnet section for each subnet on your network

NFS

- The head node is usually configured as NFS server for the home directores of the users, while the compute nodes are configured as clients
- This simplifies some operations:
  - The collection of the output at the and of the run is also facilitated
  - All files reside on the same file system
  - Backup procedures are facilitated
  - The access to the executable and the input files is facilitated

NFS server

- The file /etc/exports must be edited to specify which machines can mount which directories and how. I. e.:

```
/home   n*.grid.unipg.it(rw,no_root_squash,sync)
/opt/exp_soft *.grid.unipg.it(rw,no_root_squash)
```

- The access is granted read and write and the user mapping on an anonymous user is turned off
- Pay attention to spaces - no space must be present before the left parenthesis before the options!

NFS server

- Start the service with the command:

/sbin/service nfs start

- Query the NFS status with:

> /sbin/service nfs status
rpc.mountd (pid 4383) is running...
nfsd (pid 4377 4376 4375 4374 4373 4372 4371 4370) is running...
rpc.rquotad (pid 4366) is running...

- In some Linux distributions when restarting NFS, you may find it necessary to explicitly stop and restart the *nfslock* and *portmap* as well
- The mods in the /etc/exports file become active with the command:

/usr/sbin/exportfs -va

NFS client

- The NFS is probably already running on the nodes. You just need to mount the remote filesystem. In the long run the easiest approach is to edit the file /etc/fstab adding an entry for the server:

```
ce.grid.unipg.it:/home  /home nfs rw,defaults 0 0
ce.grid.unipg.it:/opt/exp_soft /opt/exp_soft nfs rw,defaults 0 0
```

- You can mount manually the filesystems with the mount command:

```
mount /home
mount /opt/exp_soft
```

- Keep in mind that the directories where the filesystems are mounted must already exist and that all files stored on the local system will be inaccessible after the mount: they are still there but you cannot access them.

NFS client

- If a firewall is running, it blocks the NFS traffic: in case of troubles this is the first thing to check
- User and group IDs can also create some surprises. User and Group Ids should be consistent among the systems using NFS (each user must have identical ID on all systems).
- Be aware that root privileges don't extend across NFS shared systems. So if as root you *cd* to a remotely mounted filesystem, don't expect to be able to look at every file.
- Details are available at the nfs(5), exports(5), fstab(5) and mount(8) manpages

Automount

- There's another alternative to mount filesystems (particularly important for the home directories) – using an automount program like *autofs* or *amd*
- An automount daemon mounts a remote filesystem when an attempt is made to access the filesystem and unmount the filesystem when it is no longer needed. This is transparent to the user
- Support to autofs must be compiled into the kernel before it can be used. With most Linux releases, this has already been done. If in doubt use:

```
cat /proc/filesystems
```

- In the output you should see:

```
nodev   autofs
```

Automount

- Next, you need to configure your system. Autofs uses */etc/auto.master* to determine the mount points. Each line in the file specifies a mount point and a map file that defines which filesystem will be mounted to the mount point. For example:

```
/home   auto.home   --timeout 600
```

- */home* is the mount point and *auto.home* specifies what will be mounted
- auto.home will have multiple entres such as:

```
osvaldo      ce.grid.unipg.it:/home/osvaldo
```

- NFS should be running and you may need to update your */etc/exports* file

Other cluster file system

- NFS has some limitations. First, there are some potential security issues. If you are going to use NFS it is important that you use the updated version, apply any needed patches, and configure it correctly.
- It do not scale well, so for large clusters (>1000 nodes) NFS is inadequate.
- NFS is not meant to be a high-performance, parallel filesystem. PVFS is a reasonable solution in such case
- Some technologies like Starage Area Network (SAN) and iSCSI (SCSI over IP) are interesting solutions to look at.

SSH

- To run software across a cluster, you'll need some mechanism to start processes on each machine.
- In practice, a prerequisite is the ability to log onto each machine within the cluster, without passwords.
- SSH provides mechanisms to log onto remote machines, run programs on remote machines, and copy files among machines.
- OpenSSH is the open source version of the program (http://www.openssh.org)

SSH

- There are 2 sets of the SSH protocols, SSH-1 and SSH-2. Unfortunately SSH-1 has a serious security vulnerability. SSH-2 is the protocol of choice.
- Check the status with the command:

/sbin/service sshd status

sshd (pid 16711 16574 4249) is running...

- In some distributions only the client is installed. To check the packages installed:

```
# rpm -qa |grep ssh
openssh-clients-4.3p2-4.cern
openssh-4.3p2-4.cern
openssh-server-4.3p2-4.cern
osiopenssh-VDT1.2.6rhas_3-2
```

SSH

- If needed, you can install the server from the RPM, then start the process:

```
/sbin/service sshd start
Generating SSH1 RSA host key: [ OK ]
Generating SSH2 RSA host key: [ OK ]
Generating SSH2 DSA host key: [ OK ]
Starting sshd: [ OK ]
```

- Configuration files for both the server, *sshd_config* and client, *ssh_config*, can be found in */etc/ssh*, but the defaults are quite reasonable.
- To connect to a remote machine, use the command

```
ssh user@host
```

SSH

- The first time you connect to a remote machine, you'll receive a message with the remote machine's fingerprint, a string that identifies the machine.
- The fingerprint will be recorded in a list of known hosts on the local machine. SSH will compare fingerprints on subsequent logins to ensure that nothing has changed.
- You can also use SSH to execute comands on remote systems:

```
ssh -l osvaldo n01 date
```

SSH

- The system will ask a password each time a command is run. If you want to avoid this, you'll need to do some extra work. You'll need to generate a pair of authorization keys that will be used to control access and then store these in the directory *.ssh* of your home directory

```
ssh-keygen -b1024 -trsa
```

- Two keys are generated, a public and a private key. The public key is distributed to remote machines. Add the public key in each of the system you'll want to log onto in the file *.ssh/authorized_keys2* of your home directory

Other services and configuration tasks

- **Apache** – While an HTTP server may be seen unnecessary in a cluster environment, several cluster management tools use HTTP to display results. If you want to do remote monitoring, you'll need to install an HTTP server.
- **NTP** – The Network Time Protocol is an important tool to synchronize clocks in the cluster. This is particularly important in Grid environments.
  - The file /etc/ntp.conf must contain the IP address of valid NTP servers
  - *pool.ntp.org* is a cluster of public NTP servers

Other services and configuration tasks

- **VNC** – Virtual Network Computing is a nice package that allows remote graphical login to your system. Is available as a RedHat package or from http://www.realvnc.com . It can be tunneled using SSH for greater security.
- **Multicasting** – Several clustering utilities use multicasting to distribute data among nodes within a cluster, either for cloning systems or when monitoring systems. In some cases multicasting can increase performances. It has to be enabled in the kernel.

Other services and configuration tasks

- **Name service** - Defining properly the name service (/etc/hosts, /etc/resolv.conf, DNS) will make the life easier
  - A DNS primary for the cluster resources could be recommended.At least a DNS cache only has to be configured
- **NIS** - The YP service will implement a one-time login environment. The head node is usually the NIS server, while the nodes are configured as client. This way the head node has to be started before the compute nodes

Other services and configuration tasks

- **Name service** - Defining properly the name service (/etc/hosts, /etc/resolv.conf, DNS) will make the life easier
  - A DNS primary for the cluster resources could be recommended.At least a DNS cache only has to be configured
- **NIS** - The YP service will implement a one-time login environment. The head node is usually the NIS server, while the nodes are configured as client. This way the head node has to be started before the compute nodes

Cluster Security

- The security of your cluster must be implemented at least at the following level:
  - Firewall (external to the head node)
    - Open only the used ports
  - Limit the services to the essential on the ompute nodes
  - If interactive application aren't an issue, put the compute nodes in a private network
  - Disable the root access on every machine
  - Install the security patches
  - Thest the open ports with *nmap*

openMosix

From `http://openmosix.sourceforge.net` :

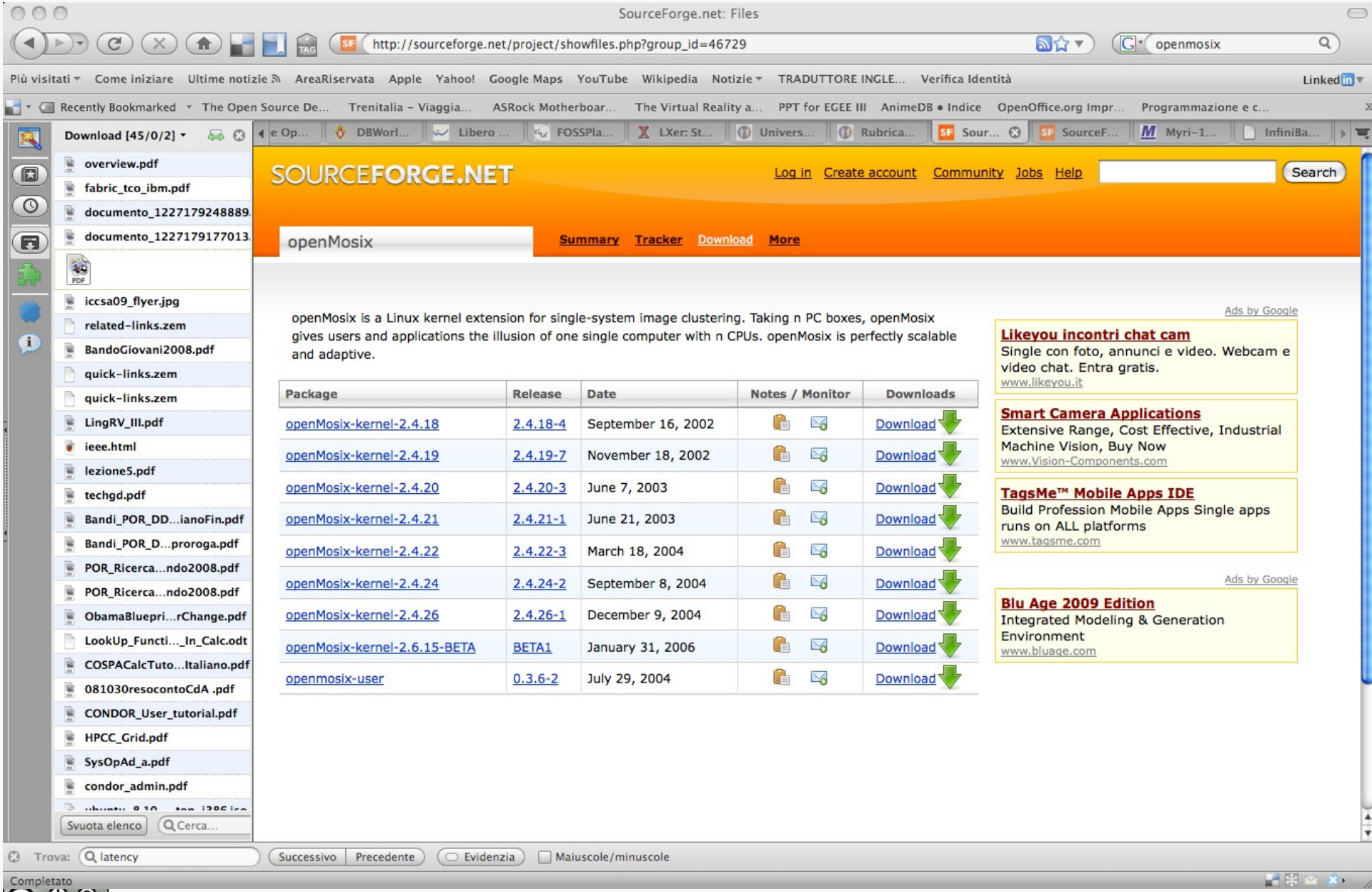
openMosix is a Linux kernel extension for single-system image clustering which turns a network of ordinary computers into a supercomputer.

The openMosix Project has **officially closed as of March 1, 2008**.

Source code and mail archives continue to be available from SourceForge as read-only resources.

Moshe Bar

Latest source files available

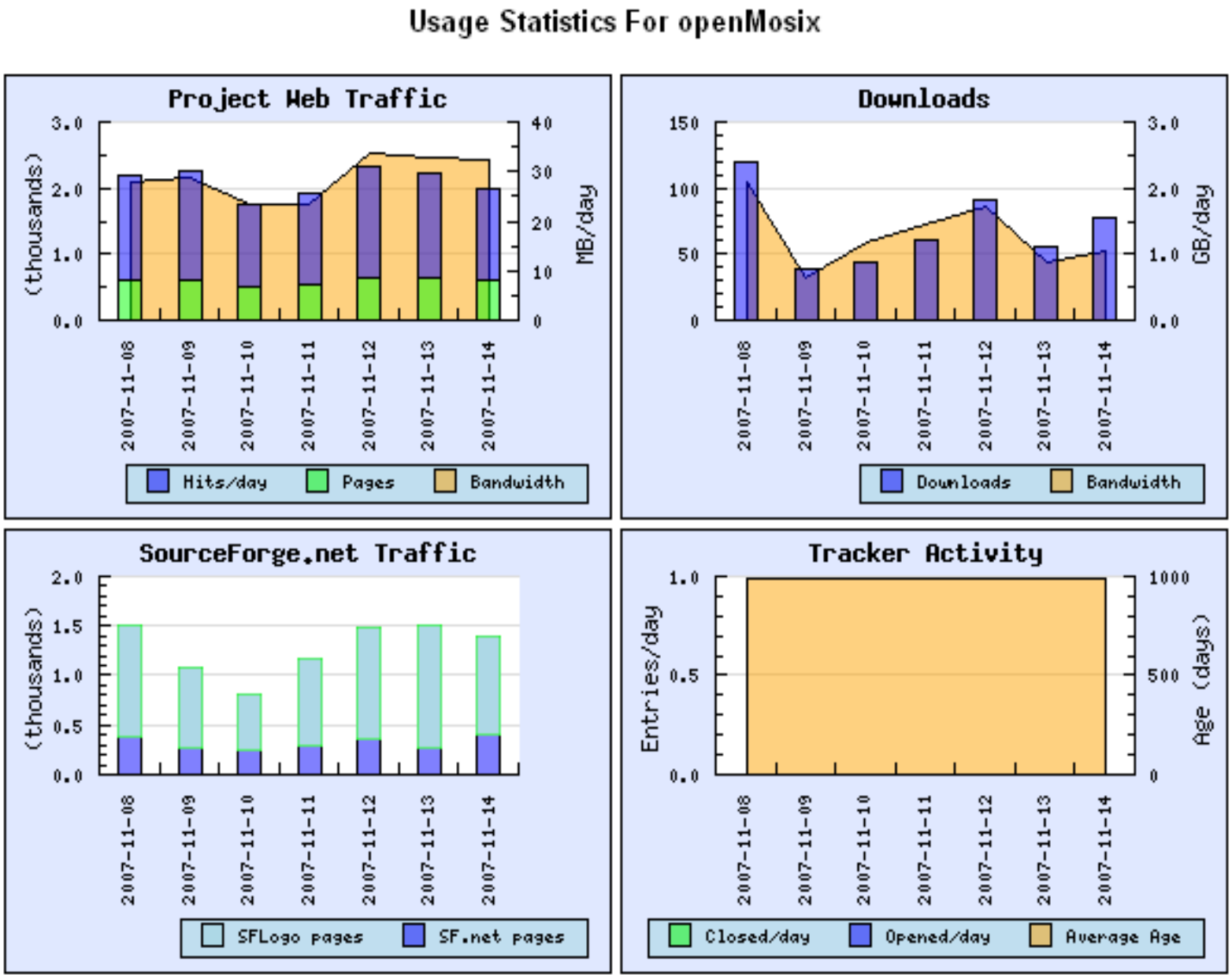


openMosix

- openMosix is a software that extends the Linux kernel so that processes can migrate transparently among the different machines within a cluster in order to more evenly distribute the workload.
- It includes both a set of kernel patches and support tools to control the migration of processes among machines.
- The process migration among machines is transparent to the user and can be controlled using the provided openMosix (and also third-party) tools

openMosix

- openMosix originates from a fork from the earlier MOSIX (Multicomputer Operating System for Unix), when MOSIX moved away from a General Public Licence.
- OpenMosix is the work of Moshe Bar, originally a member of the MOSIX team, and a number of volunteers.
- Moshe Bar announced the openMosix Project Dead of Live om March 1, 2008.
- openMosix source will remain available indefinitely on SourgeForge, frozen as of March 1, 2008



Project activity on SourceForge.Net

How openMosix works: SSI clustering

- OpenMosix is able to transform a cluster on a SSI computer, a virtual SMP machine in which each node provides a CPU.
- The communication overhead is of course high.
- The granularity for openMosix is the process.
- The rapid development of multicore SMP computers determined the end of the project even if the project activity is very high and a large number of users is downloading it.

How openMosix works

- Not all processes migrate.
  - Short time lasted (<5 sec)
  - Shared writable memory, such as web servers
  - Direct manipulation of I/O devices
  - Real-time scheduling
- If a process already migrated attemps to do any this things, the process will migrate back to its unique home node (UHN)
- To support process migration, openMosix divides processes into two parts:
  - User contexts (program code, stack, data, etc)
  - System context (resources attached to, kernel stack) [daes not migrate]

How openMosix works

- openMosix uses an adaptive resource allocation policy (each node monitors and compares its own load with the load of the other computers)
- When a more lightly loaded computer is found, the attempt of migration is made.
- As the load of individual computers change, processes will migrate among the computers to rebalance the load dinamically
- Individual nodes, acting as autonomous systems, decide which process migrate

How openMosix works

- The communication among small sets of nodes within the cluster used to compare loads is randomized (clusters scale well because of this random element)
- Since communications is within subsets in the cluster, nodes have limited but recent information about the state of the whole cluster -> reduces overhead and communication
- OpenMosix API uses the value in the flat files /proc/hpc to record and control the state of the cluster

How openMosix works

- While load comparison and process migration are automatic within a cluster, openMosix provides tools to control migration.
  - alter the cluster's perception of how heavily an individual node is loaded
  - Tie processes to a specific computer
  - Block the migration of a process to a computer

Group migration

- OpenMosix allows to migrate group of processes (small kernel patch that creates files /proc/[PID]/miggroup)

- make processes join a miggroup:
[root@node1 miggroup]# ./joingroup 1300 1332 1364 1396 1428 1460 1
miggroup = 1
processes = 1300 1332 1364 1396 1428 1460
process 1300 joins the miggroup 1 now
process 1332 joins the miggroup 1 now
....
- show all processes from a specific miggroup:
[root@node1 miggroup]# ./showgroup 1
1300 with commandline bash belongs to 1
1332 with commandline bash belongs to 1
....

- migrate a miggroup to another node:
[root@node1 miggroup]# ./migrategroup 1 2
migrate 1300 to 2
migrate 1332 to 2
migrate 1364 to 2

- reset a specific miggroup:
[root@node1 miggroup]# ./resetgroup 1
resetet 1300
resetet 1332

Installation

- Since openMosix is a Kernel extension it won't work with just any kernel (currently the version 2.4-26-1 IA32-compatible Linux kernel is supported). An IA64 port is also available
- Among others openMosix has been reported to work on Debian, Gentoo, RedHat, and SUSE Linux.
- Knoppix, BCCD and PlumpOS, three bootable CD Linux distributions, include openMosix
- To build an openMosix cluster you need to install the extended openMosix kernel in each of the nodes of the cluster.

Installation

- If you are using a suitable Linux distribution and you don't have special needs, you may download a precompiled version of the kernel
- Otherwise you need a clean copy of the kernel source, apply the openMosix patches, recompile the sources, and install the patched kernel.
- While using a precompiled vesion of the kernel is the easiest way to go, it has a few limitations.
- The documentation is a little weak with the precompiled kernel, so you won't know exactly what options have been compiled (however the *.config* files are available via CVS)

Installation

- The openMosix user tools should be downloaded when you download the openMosix kernel patches
- Additionally you will also want to download and install openMosixView, third party tools for openMosix
- To install a precompiled kernel you need to download (http://openmosix.sourgeforge.net) the appropriate files and packages, installing them and making a few minor configuration changes
- You must create an emergency boot disk if you don't have one: you are adding a new kernel...

Configuration changes

- While the installation will take care of the stuff that can be automated, there are a few changes you'll have to do manually to get openMosix running.
- The next time you reboot your system, openMosix won't be the default kernel for the loader. If you are using GRUB, then you'll need to edit /etc/grub.conf to select the openMosix kernel (default=0). If you are using LILO the procedure is pretty much the same, except that you'll need to manually create the entry in the /etc/lilo.conf file (default=openMosix) and rerun the loader /sbin/lilo -v

Configuration changes

- Another issue is wheter your firewall will block openMosix traffic. openMosix uses the port ranges: UDP 5000-5700, UDP 5428, TCP 723 and 4660. You will need to allow any other related traffic (SSH, NFS, etc)
- openMosix needs to know about the other machines in your cluster. You can use the omdiscd tool or edit manually the /etc/openmosix.map
- Routing must be correctly configured for omdiscd to run correctly.

/etc/openmosix.map

- Its simplest form has one entry per each machine. Each entry consists of three fields: a unique device node number(starting at 1), the machine's IP address (or names defined in the /etc/hosts), and a 1 indicating that it is a single machine.
- It is possible to have a single entry for a range of machines that have contiguous IP addresses. In that case, the first two fields are the same. The third field is the number of machines in the range.
- The map file must be replicated in all nodes
- You can list the map that openMosix is using with the showmap command

User tools

- During the installation a script called openmosix is copied into /etc/init.d so that openMosix will be started automatically at boot.
- You can check your installation with /etc/init.d/openmosix status
- At its simplest openMosix is transparent to the user. You can sit back and reap the benefits.
- If you want more control you need to install openMosix user tools that contain several useful management tools (migrate, mosctl, mosmo, mosrun and setpe), mps and mtop

mps and mtop

- Both commands look a lot like their counterparts ps and top. The major difference is that each has an additional column that gives the node number in which a process is running

```
# mps
PID TTY NODE STAT TIME COMMAND
...
19766 ? 0 R 2:32 ./loop
19767 ? 2 S 1:45 ./loop
19768 ? 5 S 3:09 ./loop
19769 ? 4 S 2:58 ./loop
19770 ? 2 S 1:47 ./loop
19771 ? 3 S 2:59 ./loop
19772 ? 6 S 1:43 ./loop
19773 ? 0 R 1:59 ./loop
```

migrate

- The tool migrate explicitly moves a process from one node to another.
- Since there are circumstances under which some processes can't migrate, the system may be forced to ignore this command
- You'll need the PID and the node number of the destination machine

```
migrate 19769 5
```

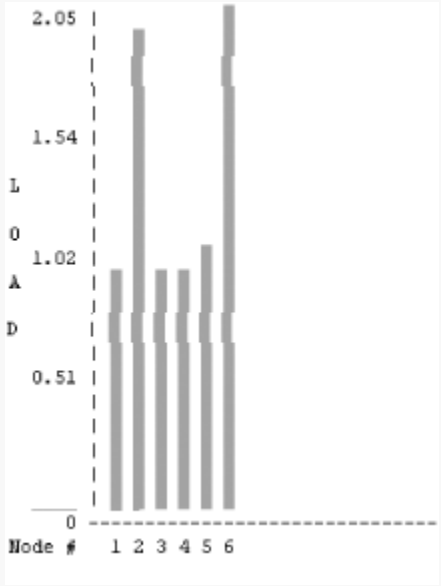
- This command migrates process 19769 to node number 5 (you can use home in place of the node number to send a process back to the CPU where it was started)

mosctl

- With mosctl you have greater control over how processes are run on individual machines.
- The speed option overrides the node's idea of its own speed. This can be used to attract or discourage process migration to the machine.
- It can be used to display the utilization or tune the performance parameters.
- The command has too much options to be described in detail (see the manpage)

mosmon

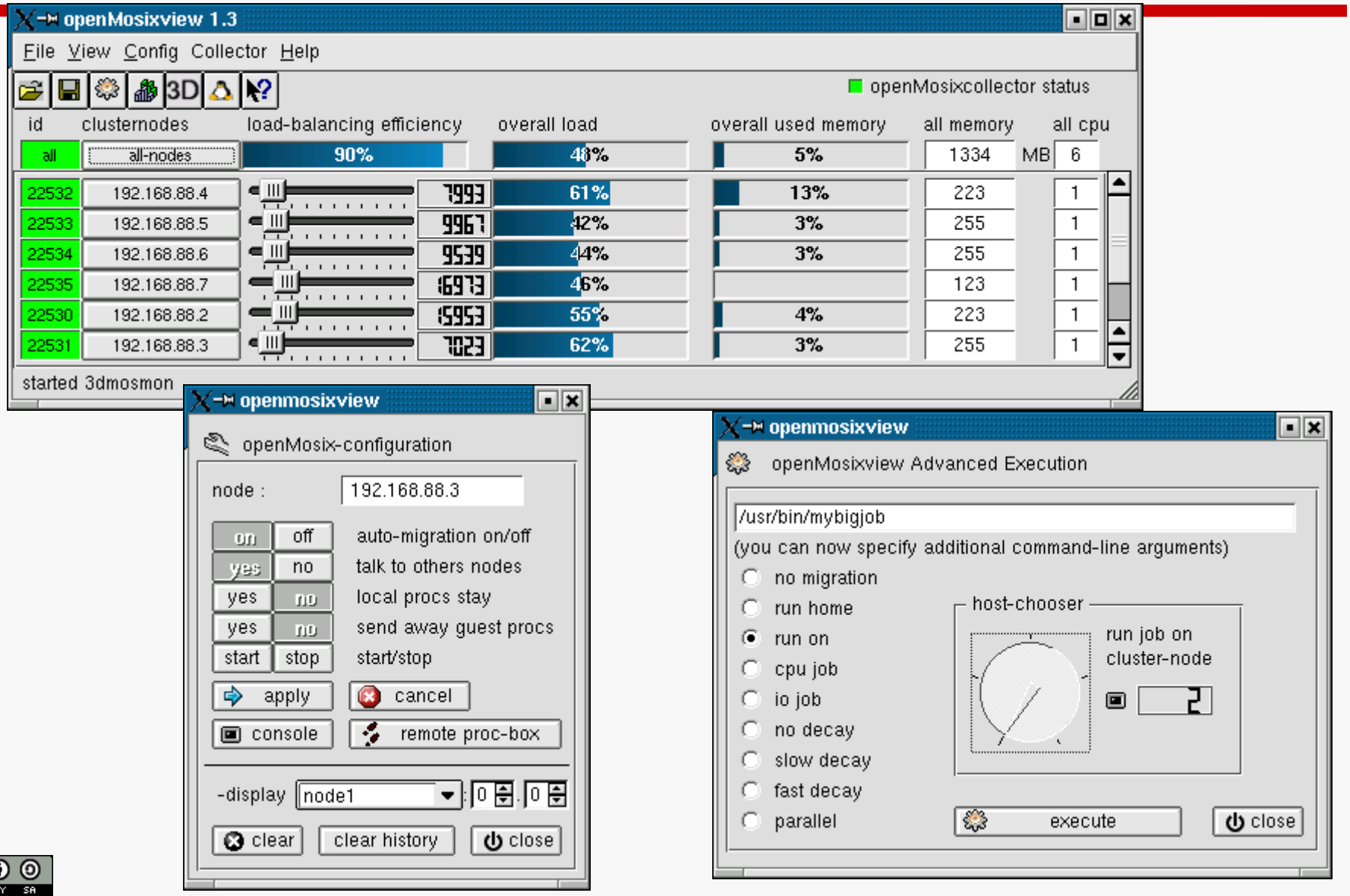
- Mosmon utility gives a good idea of what is going across the cluster.
- It is an ncurses-based utility that will display a simple bar graph showing the loads on the nodes



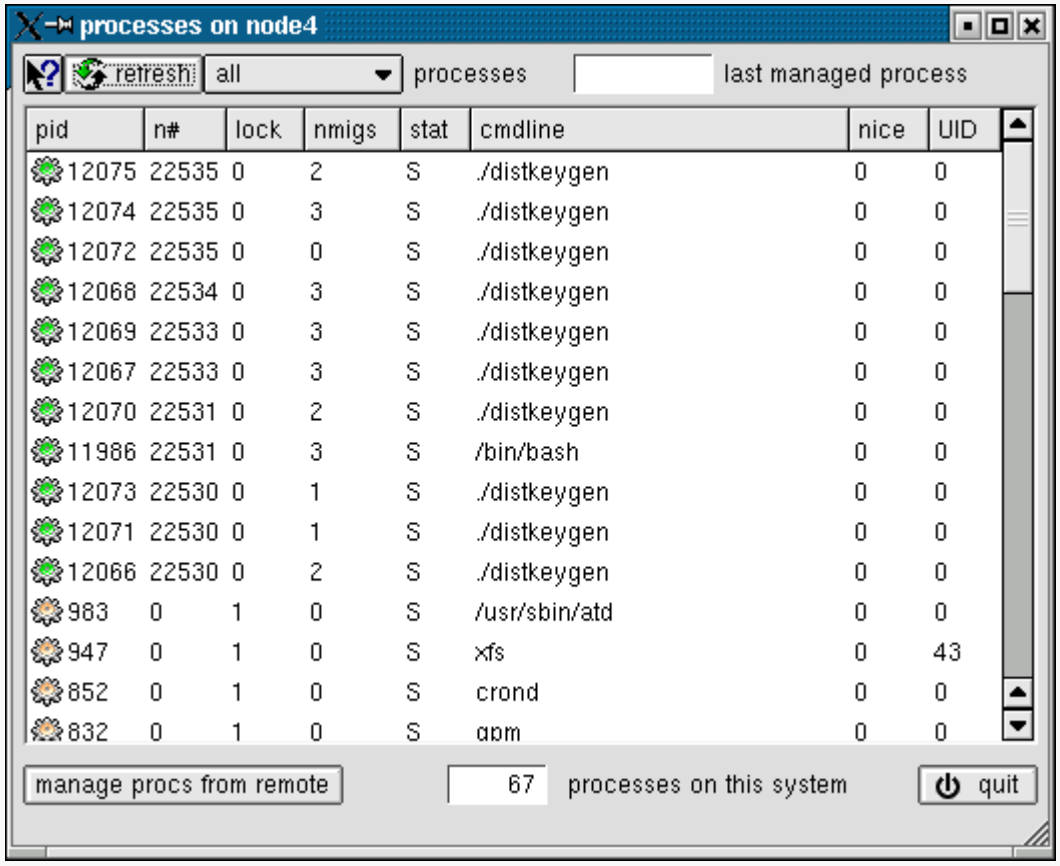
mosrun

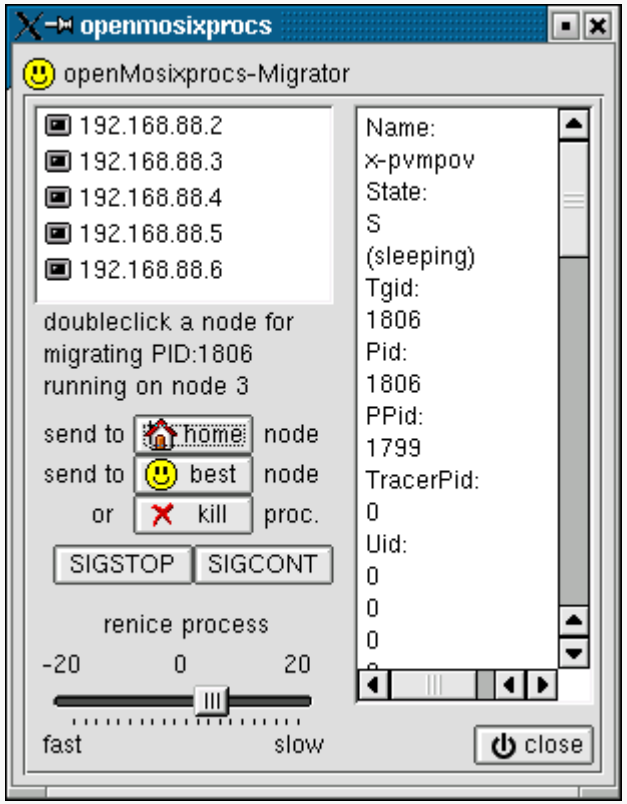
- **mosrun** command can be used to advise the system to run a specific program on a specified node.
- You'll need the program name and the destination node number (use -h to address the home node)
- **setpe** command can be used to manually configure a node (it is used by `/etc/init.d/openmosix` rather than used directly).
- You can use it to start/stop openMosix:
  #/sbin/setpe -w -f /etc/openmosix.map
- Useful options: **-z** to read the configuration file, **-c** to check the map's consistency, **-off** to shutdown

openMosixView

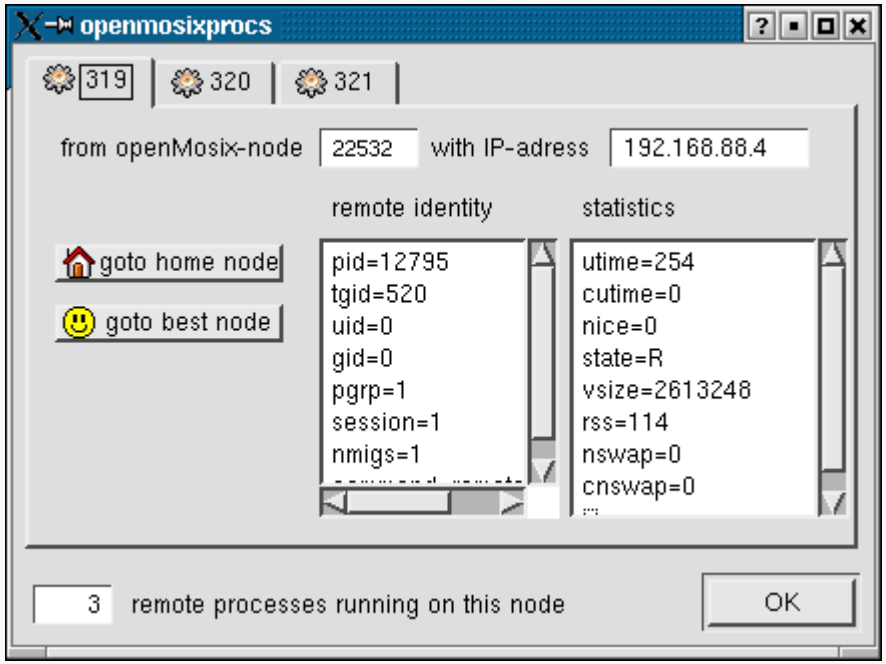


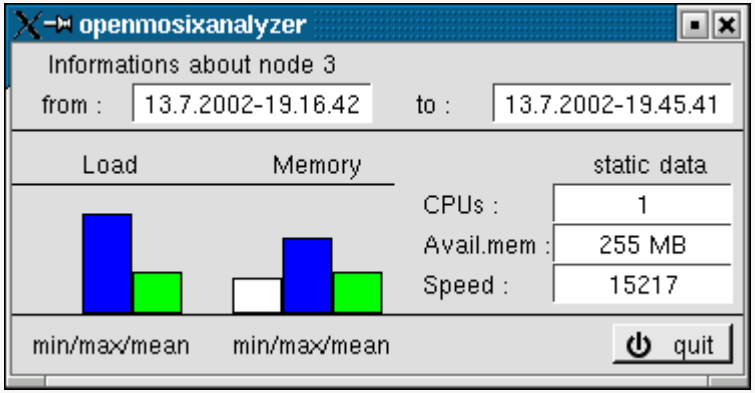
openMosixView



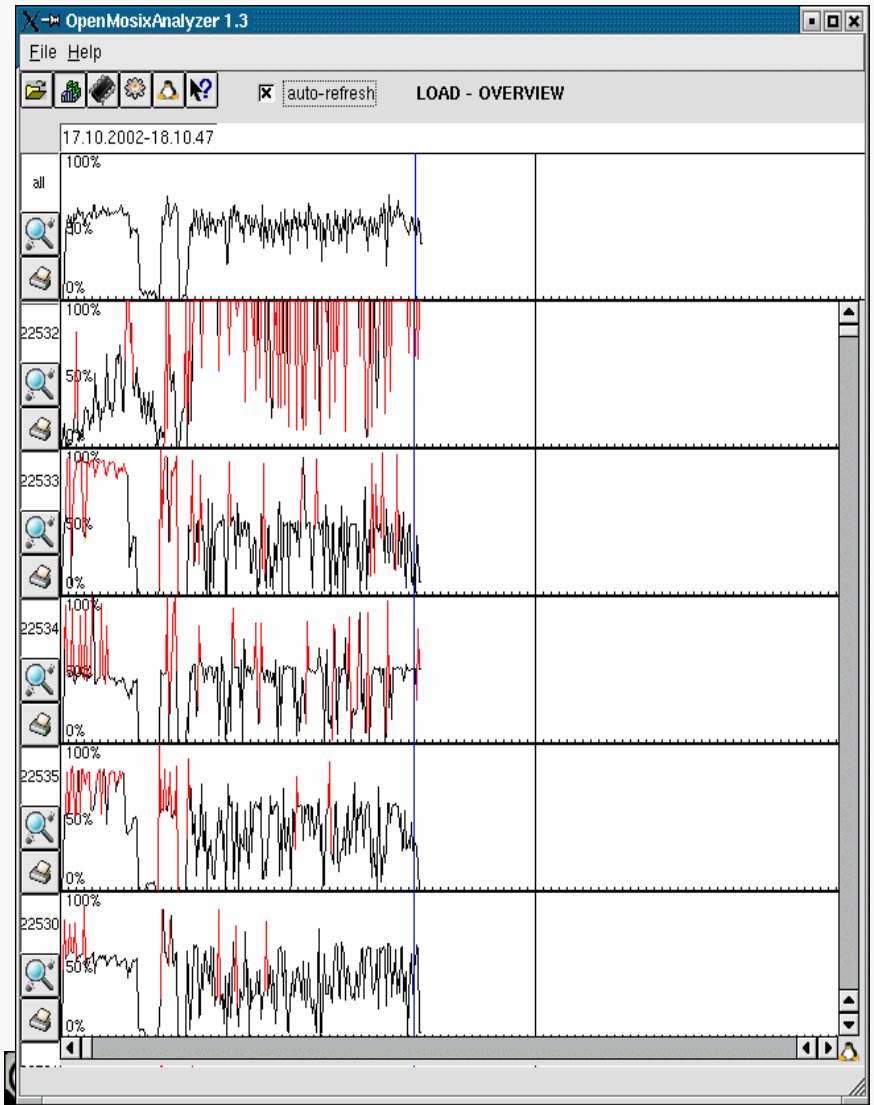


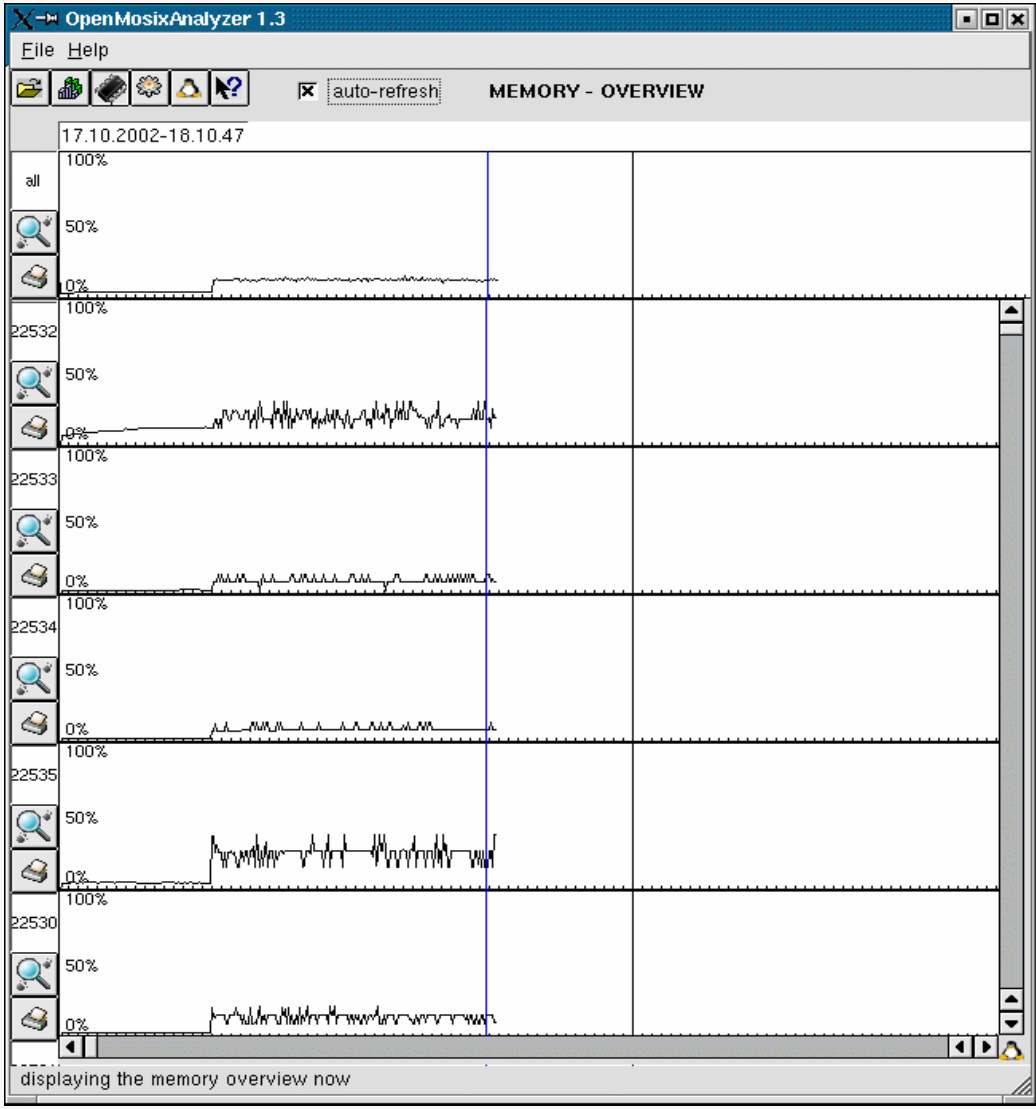
openMosixView





openMosixView





OSCAR


open source cluster application resources
OSCAR
Blog
View Tickets
Browse Source
Roadmap
Timeline
Search
Wiki
Start Page
Index
History
Last Change
Main Page
Current Status
Download
Documentations
Contact
Cluster Register
Search:
What is OSCAR ?
OSCAR allows users, regardless of their experience level with a *nix environment, to install a Beowulf type high performance computing cluster. It also contains everything needed to administer and program this type of HPC cluster. OSCAR's flexible package management system has a rich set of pre-packaged applications and utilities which means you can get up and running without laboriously installing and configuring complex cluster administration and communication packages. It also lets administrators create customized packages for any kind of distributed application or utility, and to distribute those packages from an online package repository, either on or off site.
OSCAR installs on top of a standard installation of a supported Linux distribution. It installs and configures all required software for the selected packages according to user input. Then it creates customized disk images which are used to provision the computational nodes in the cluster with all the client software and administrative tools needed for immediate use. OSCAR also includes a robust and extensible testing architecture, ensuring that the cluster setup you have chosen is ready for production.
The default OSCAR setup is generally used for scientific computing using a message passing interface (MPI) implementation, several of which are included in the default OSCAR package set. One of OSCAR's strengths is that it is possible to install multiple MPI implementations on one cluster and switch easily between them, either at the system default level or the individual user level.
Other types of applications which use clusters of computers, such as load balancing web clusters and high availability clustering packages, would certainly be fairly easy to implement using the OSCAR package system but are outside the expertise of our current development team.
Anyone is welcome to contribute to OSCAR core development, or to submit packages to be included in the default OSCAR repositories. We are a community driven project and are always on the lookout for new talent and ideas.
How can I contribute?
Email osl-sysadmin@ osl . iu . edu or post a message to oscar-devel@... to get your subversion and trac account. We will send an invitation mail to let you create your own account for our systems(i.e., subversion and trac).
Latest release
On February 8, 2011: OSCAR 6.1.0
On April 8, 2010: OSCAR 6.0.5
On September 25, 2009: OSCAR 6.0.4

The OSCAR package

- Setting up a cluster can involve the installation and configuration of a lot of software as well as reconfiguration of the system and previously installed software.
- The OSCAR (Open Source Cluster Application Resources) is a software package that is designed to simplify cluster installation.
- OSCAR includes everything that you are likely to need for a dedicated high-performance cluster.
- OSCAR takes you completely through the installation of your cluster

The OSCAR package

- The design goal for OSCAR include using the best-of-class software, eliminating the downloading , installation and configuration of individual components and moving towards the standardization of clusters
- OSCAR was created and is maintained by the Open Cluster Group.
- It is designed for the High Performance Computing and for asymmetric clusters.
- Actually OSCAR could be used for any cluster application (see HA-OSCAR group)

Custom installation of packages

- It is possible tu customize the packages included in the OSCAR distribution.
- Although OSCAR does not provide a simple mechanism to do post-installation configuration of the added packages, it is possible to include some scripts to configure them from the installation procedure (see C3 tool).
- Sometimes the included packages aren't the most updated ones, because the integration of the variety of individual packages is the driving approach.

Custom installation of packages

- OSCAR brings together a number of software packages for clustering. There are some unique OSCAR scripts available in the distribution.
- OSCAR provides a script, the Oscar Package Downloader (opd) that simplifies the download and installation of additional packages that are available form OSCAR repositories in an OSCAR compatible format.
- Opd is so easy to use that any package available through opd can be considered part of OSCAR.
- Opd can be invoked as a standalone program or from the OSCAR installation wizard.

OSCAR packages

- Core packages (must be installed)
  - Core
  - C3 (Cluster, Command and Control tool)
  - Environment Switcher
  - oda (Oscar Database Application)
  - perl-qt (Perl OO interface to Qt GUI toolkit)
  - SIS (System Installation Suite)

OSCAR packages

- Included packages (are distributed as part of OSCAR, but can opt out on installing them)n
  - Disable-services - this script disables unneeded programs on the clients(kidzu, slocate, mail services)
  - networking – the cluster head is configured as cache DNS for the clients
  - ntpconfig
  - kernel_picker – used to change the kernel used in your SIS image before building the cluster nodes
  - loghost – configures the syslog settings

OSCAR packages

- Additional packages (third-party packages available for download and compatible with OSCAR – not required)
  - Autoupdate
  - Clumon (by opd)
  - Ganglia (by opd)
  - MAUI
  - Myrinet drivers
  - OpenPBS
  - Pfilter
  - PVFS (by opd)
  - OPIUM
  - Torque (by opd)
  - VMI (by opd)

Reconfigured services

- OSCAR will install and configure (or reconfigure) a number of services and packages of Linux:
  - Apache
  - DHCP
  - NFS
  - mySQL
  - openSSL
  - openSSH
  - rrdtool
  - python
  - rsync
  - php

Programming tools

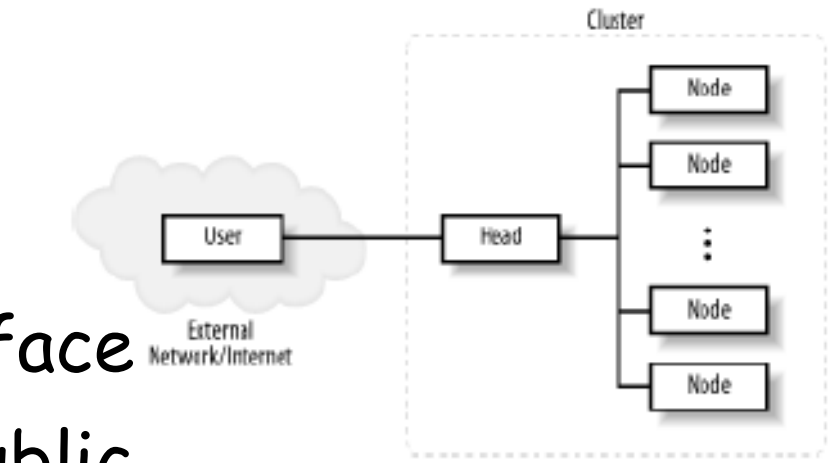
- HDF5 – Hierarchical Data Format library for scientific data
- LAM/MPI – implementation of the message passing interface (MPI) libraries
- MPICH – implementation of the message passing interface (MPI) libraries
- MPICH-GM (by opd) – provides MPICH with support for low-level message passing for Myrinet networks
- MPICH-VMI (by opd) – implementation of MPICH that uses VMI
- PVM – Parallel Virtual Machine libraries (message passing)

Installing OSCAR

- Because OSCAR is a complex set of hardware that includes a large number of programs and services , it can be very unforgiving if you make mistakes when setting it up.
- For some errors you have to start from scratch
- First you need to plan your system
- Install OSCAR on the cluster's head node
- It is recommended that you begin with a clean install of your operating system and that you customize your OSCAR installation as little as possible the first time you install it.

Network configuration

- It is common that the head node is dual homed.
- OSCAR will set up a DHCP server on the private interface
- The configuration of the public interface will be determined by the configuration of the external network.
- For the internal network is recommended to use a private address space



Network configuration

- Once you have selected the address space, you may configure the private interface using a tool of your choice (neat, ifconfig or netcfg)
- You will need to set the IP address, subnet mask and default gateway. You have to configure the interface to be active at the startup and the packet forwarding.
- Reboot the machine and verify the configuration (mainly that the packets are fowarded to the internal network).

Network configuration

- If you have the security set too tightly on the server, it will interfere with the client installation
- Turn off, if active, the Security Enhanced Linux (SELinux).
- Verify that the firewall is working properly (if already installed, check the SSH functionality, in particular that PermitRootLogin is set to yes)
- Verify that the interface names are correct
- This is a good point to do other basic configuration tasks, such as setting up printers, setting the message of the day, etc.

Loading software on the server

- The next step is to get the software you'll need onto the server. This consists of the OSCAR distribution and the Linux packages you need to build the image for the client machines
- First create the directory /tftpboot/rpm and then copy over the packages
- The Linux packages will be coped from the installation CDs
- You can download the OSCAR package from http://oscar.sourceforge.net
- You'll have the option of downloading OSCAR with or without sources (SRPMs).

Installing the OSCAR package

- Next you will unpack OSCAR code in a suitable directory (/opt)
- Rename the directory "oscar-5.0" to "oscar"
- Be sure the directory "/tftpboot" exists
- Create the /tftpboot/oscar and the /tftpboot/distro directories
- Move the oscar-repo-common-rpms and the oscar-DISTR-VER-ARCH tarballs suitable for distribution in the /tftpboot/oscar and unpack them

Advanced repository creation

- OSCAR also needs to have access to the distribution packages in order to resolve dependancies on the master node and to be able to build the client node images
- You must prepare the package repository, copying the rpms files from the distribution packages to the package repository directory
- From OSCAR 5.0 the package repository structure has been redesigned and split up such that multiple distributions and architecture can be supported

/tftpboot/distro/$DISTRIB-$VERS-$ARCH

Supported distrib. Name and version examples

distro name	version	architecture	repository path
RedHat Enterprise Linux WS	4	i386	/tftpboot/distro/redhat-el-ws-4-i386
Fedora Core	4	x86_64	/tftpboot/distro/fedora-4-x86_64
CentOS	4.3	ia64	/tftpboot/distro/centos-4-ia64
Scientific Linux	3.6	i386	/tftpboot/distro/scientificlinux-3-i386
Mandriva Linux	2006	i586	/tftpboot/distro/mandriva-2006-i386

- Remote Url repositories are specified in the file

  /tftpboot/distro/$DISTRIB-$VERS-$ARCH.url

Launch the installer

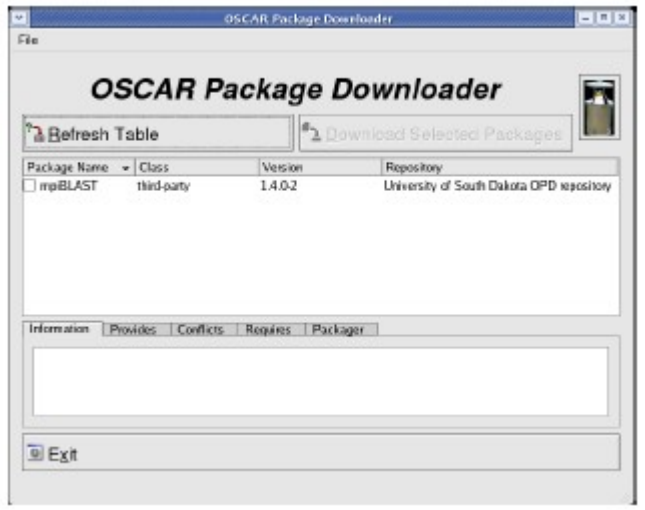
- Cd to the /opt/oscar dir and execute the command

  ./install cluster <device>
- Where <device> represents the private network adapter.
- The script will
  - Install prerequisite packages on the server
  - Install all OSCAR server RPMs
  - Update /etc/hosts with OSCAR aliases
  - Updates /etc/exports
  - Adds OSCAR paths to /etc/profile
  - Updates system startup scripts
  - Restart affected services

OSCAR wizard

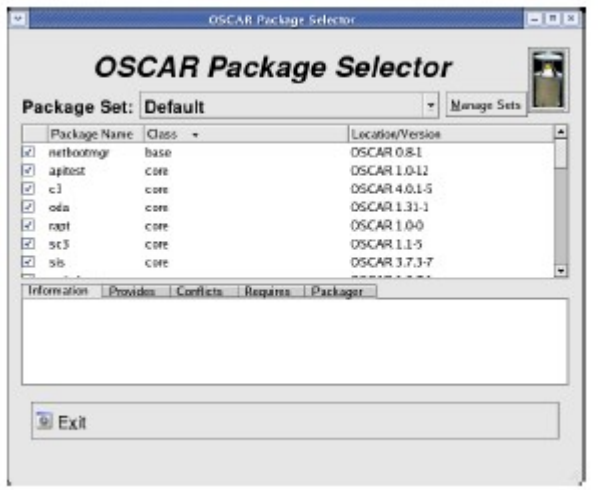
OSCAR Package Downloader

- The command line usage:
  cd $OSCAR_HOME
  ./scripts/opd
- opd uses wget, if found
- Opd will make use of the "http_proxy" environment variable, if it is set.



OSCAR Package Selector

- The window only shows OSCAR packages, it doesn't show individual RPMs
- After selected the package click on "Exit" to save the selection and return to the main OSCAR window
- There is no way of defaulting to the original settings



Configuring OSCAR Packages

- This step is optional
- If you don't click the button, all packages will be used
- Selecting a default MPI implementation
  - Altough multiple MPI implementationscan be installed, only one can be "active" for each user at a time
  - The Environment Switcher package provides a convenient mechanism for switching between multiple MPI implementations.

Install OSCAR Server Packages

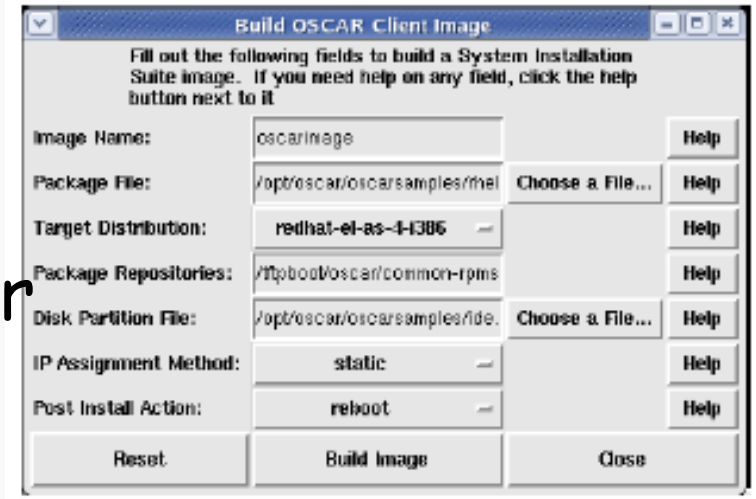
- This is the first required step
- This will invoke the installation of various RPMs and auxiliary configuration on the server
- The execution may take several minutes; text output and status messages will appear in the shell window.
- A popup will appear indicating the success or failure of this step.

Build OSCAR Client Image

- To execute this step ensure that the following conditions on the server are true:
  - During the installation the SSH daemon option (in /etc/ssh/sshd.conf) PermitRootLogin must be set to yes because this file is copied to the clients and the head node needs the root access on the nodes. After the installation this option may be set to no.
  - The TCP wrapper settings must be "not too tight", The /etc/hosts.allow and /etc/hosts.deny files should allow all traffic in the private subnet
  - Beware of firewall software that restricts traffic in the private subnet.

Build OSCAR Client Image

- A dialog will be displayed; in most cases the defaults are fine.
- You have to check that the disk partition file is the proper type for the client node.
- You may also change the postistallation action and the IP assignement method
- It is important to check BIOS settings on the client nodes



Disk partitioning

The partitions on clients are described by the disk partition file using the following format:

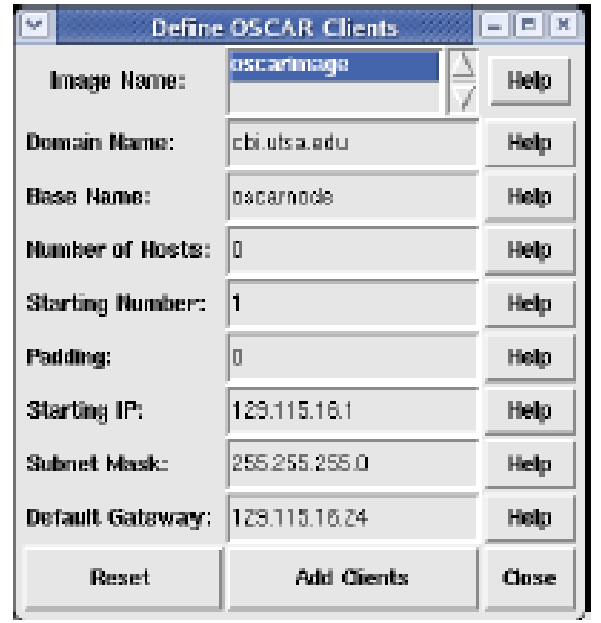
<partition> <size in MB> <type> <mount point> <opt>

Here is a sample:

```
/dev/sda1 24 ext3 /boot defaults
/dev/sda5 128 swap
/dev/sda6 * ext3 / defaults
nfs_oscar:/home - nfs /home rw
```

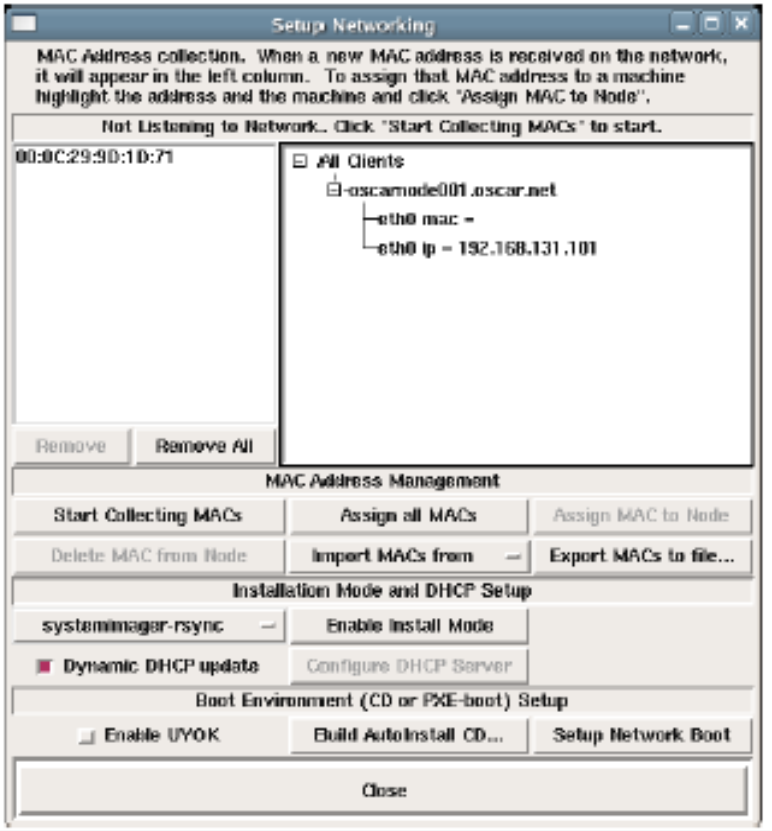
Define OSCAR Clients

- The dialog box requires few informations (image file to be used, base name to build the client names, number of nodes, starting number used to derive the client names, padding characters to fill the client name, starting IP. Subnet mask, default gateway)



Setup networking for Clients

- If you need to collect the MAC addresses of client nodes you need to enter this option of the menu



Select installation mode + Setup Boot environment

- SystemImage is the tool for OSCAR To deploy the images to cluster nodes (it is part of the System Installation Suite (SIS))
- Three transports are available:
  - Systemimager-rsync
  - Systemimager-multicast
  - Systemimager-bt
- Boot Environment:
  - Build autoinstall CD
  - Setup Network Boot

Manual setup for UYOK

- If you wish to set up the Using Your Own Kernel (UYOK) functionality by hand the following steps are required (unnecessary if the hardware of nodes and head node are similar).
  - To use UYOK to generate kernel/ramdisk pairs on the head node:

    si_prepareclient –server *servername* –no-rsyncd

    The output files will be stored in /etc/systemimager/boot
  - Now copy these files to /tftpboot if you are PXE-booting.
  - Edit your /tftpboot/pxelinux.cfg/defaultfile with a large ramdisk_size in the kernel append statement

Manual setup for UYOK

- Edit your /tftpboot/pxelinux.cfg/default file with a large ramdisk_size in the kernel append statement, eg:

```
LABEL systemimager
KERNEL kernel
APPEND vga=extended initrd=initrd.img root=/dev/ram
MONITOR_SERVER=192.168.0.2
```

- Now SystemImager will use the UYOK boot package (Which should recognise your hardware) to boot your nodes.

Client installations

- You will boot your client nodes and then they will be automatically installed and configured
- The recommended method is via network boot. The most convenient way is via PXE-boot if the network card supports it. An other option is the Etherboot project (if you can generate a boot-ROM for your network card)
- If the network cards support PXE, then change the BIOS settings such that "Network" is always first in boot order.

Client installations

- After the installation of the node, its next boot action will be automatically changed to "LocalBoot" meaning that it will boot from hard-disk
- If this will fail you'll be able to change the option via netbootmgr widget
- Then the Client nodes will be rebooted. As each machine boots, it will automatically start downloading and installing the OSCAR image from the server node.

Check completion status of the nodes

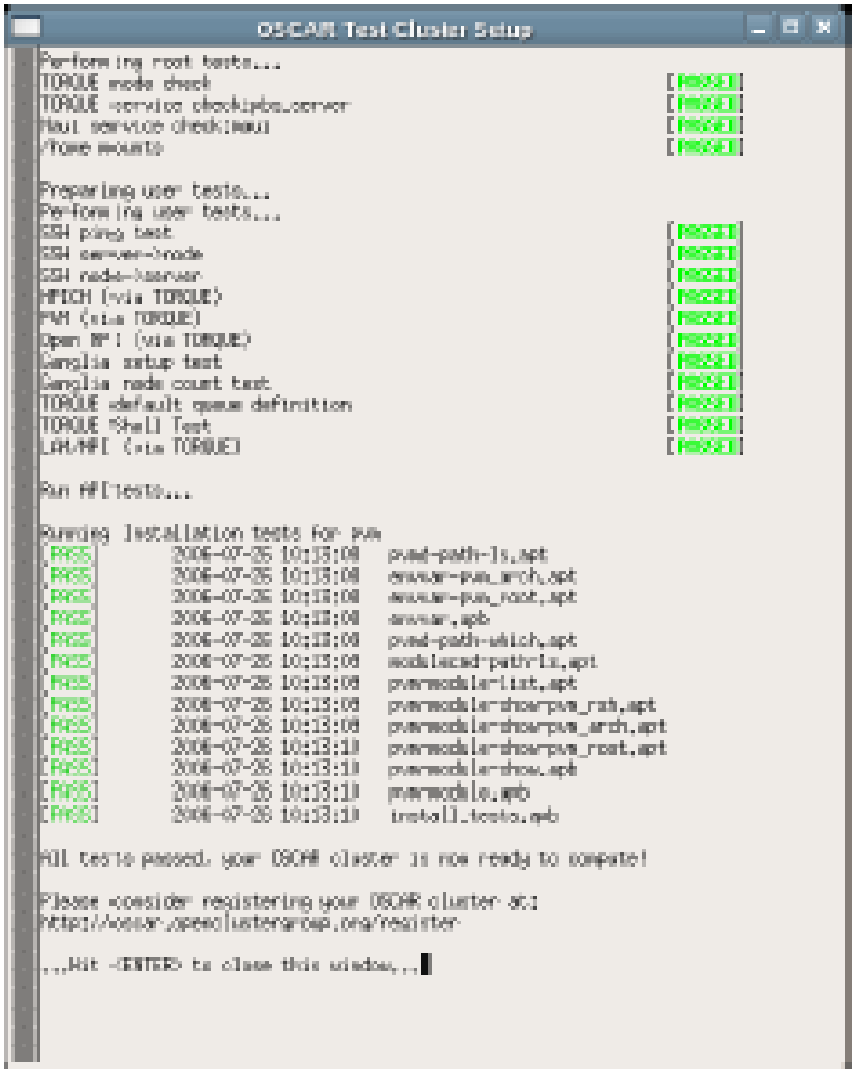
- You can check the status of the client installation using the "Monitor Cluster Deployment" functionality.
- The available post-install actions for your nodes are: halt, reboot, beep incessantly
- If the "reboot" action has been selected, you will be notified when the nodes are rebooted via the <Monitor Cluster Deployment> widget.
- After confirming that a client has completed the installation, you should reboot the node from its hard disk.

Complete the Cluster setup

- After all clients rebooted, select the above option from the menu
- During this phase several OSCAR installation and configuration scripts will be run.
- A success or failure pop-up will appear at the end of the process.

Test Cluster setup

- A simple test suite is provided in OSCAR to ensure that the key cluste components (OpenSSH, TORQUE, MPI, PVM, etc) are functioning properly
- If the test succeeds, the cluster is ready



Rocks

Rocks Clusters

http://www.rocksclusters.org/wordpress/

Rocks cluster Linux dist

Più visitati | Come iniziare | Ultime notizie | AreaRiservata | Apple | Yahoo! | Google Maps | YouTube | Wikipedia | Notizie | TRADUTTORE INGLE... | Verifica Identità | Linked

Recently Bookmarked | AnimeDB • Leggi ar... | The Open Source De... | Trenitalia – Viaggia... | ASRock Motherboar... | The Virtual Reality a... | PPT for EGEE III | AnimeDB • Indice | OpenOffice.org Impr...

Accesso... | Rock... | CS Safa... | Orari e ... | Comput... | The Op... | DBWorl... | Libero ... | FOSSPla... | LXer: St... | oscarv ... | PXE Boo... | Sou

Home | Downloads | Support and Docs | Rocks-A-Palooza | Register Your Cluster | Licensing | About

November 10, 2008

Clustercorp Releases the Mellanox OFED Roll, Intel Roll, PGI Roll, TotalView Roll, and Absoft Roll for Rocks v5.1

Tim McIntire of Clustercorp announces:

Clustercorp has released the following Rolls for Rocks v5.1 (x86_64 architecture):

- Mellanox OFED Roll

The Mellanox OFED Roll includes OFED 1.3.1. This Roll replaces the Cisco OFED Roll for Rocks 5.0 and below.

- Intel Developer Roll

The Intel Developer Roll includes C, C++, and Fortran 10.1.018, Intel Cluster Toolkit 3.1.1.003, and MKL 10.0.5.025.

- PGI Roll

The PGI Roll includes Portland Group's C, C++, and Fortran Compilers 7.2-5.

Recent News

- Rocks v5.1 Beta is Released
- Clustercorp Releases the TotalView Roll for Rocks 5
- Condor Roll for Rocks v5.0 is Released
- Service Pack Roll v5.0.3 for Rocks v5.0 is Released
- Viz Roll for Rocks v5.0 is Released
- Service Pack Roll v5.0.2 for Rocks v5.0 is Released
- NBCR Releases Opal Roll for Rocks 5.0
- Service Pack Roll v5.0.1 for Rocks v5.0 is Released

Archives

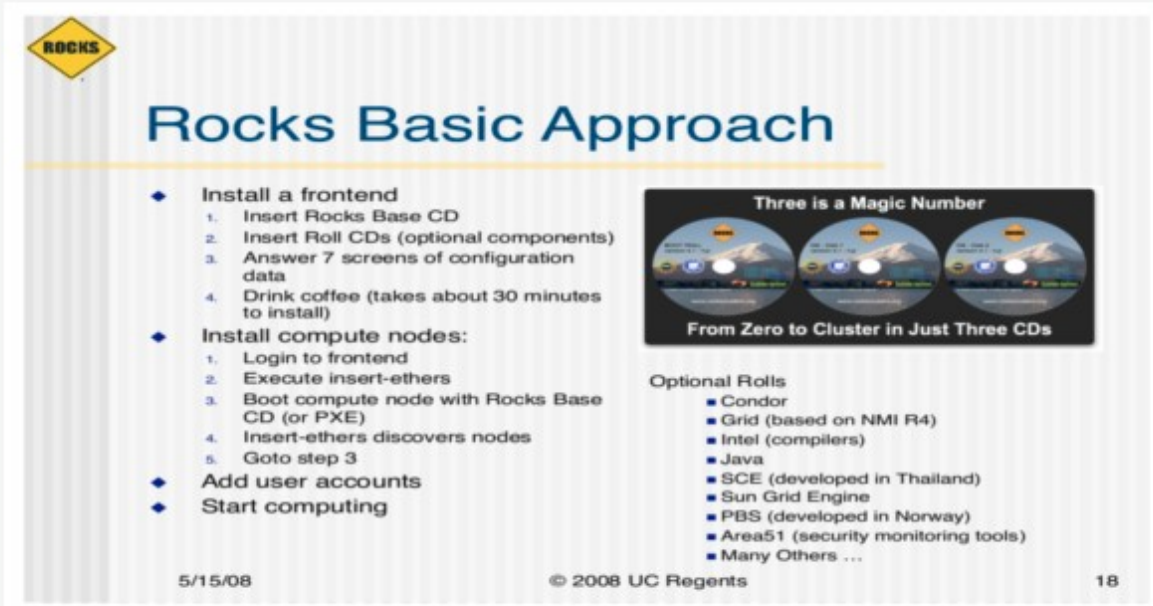
- November 2008
- September 2008
- July 2008
- June 2008
- May 2008
- April 2008

Trova: latency | Successivo | Precedente | Evidenzia | Maiuscole/minuscole

Completato

Open-Source Toolkit
for Real and Virtual Clusters

HOME | DOWNLOADS » | SUPPORT AND DOCS » | REGISTER YOUR CLUSTER | ABOUT » | Search



ROCKS TUTORIALS

Some Selected Tutorial Presentations Basic Overview of Rocks Clusters **More »**

Service Pack Roll for Rocks 6.1 Released

MARCH 11, 2013 ADMIN

The Service Pack Roll for Rocks 6.1 is now released.

This fixes a number bugs/errors/omissions in Rocks 6.1 (Emerald Boa). All new clusters should be built

READERS COMMENTS

Recent Posts | Comments | Pages

Service Pack Roll for Rocks 6.1 Released
March 11, 2013 By admin

Rocks 6.1 (Emerald Boa) for i386 and x86_64 Released!
November 29, 2012 By admin

Rocks 6.1 (x86-64) Beta (Emerald Boa) Available
October 23, 2012 By admin

Rocks Mamba (5.5/6.0) is Released
May 09, 2012 By admin

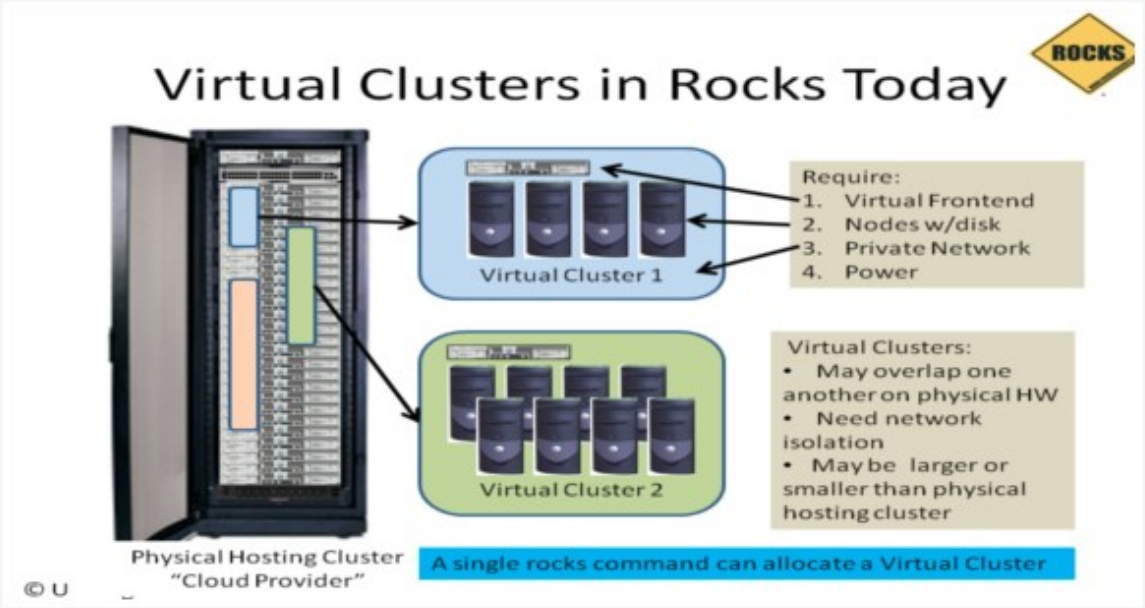
Rocks Version 5.5/6.0 (Mamba) Beta Slated for End of March 2012
March 17, 2012 By admin

MONTHLY ARCHIVE

Select Month

Open-Source Toolkit
for Real and Virtual Clusters

HOME | DOWNLOADS » | SUPPORT AND DOCS » | REGISTER YOUR CLUSTER | ABOUT » | Search



SELECTED PRESENTATIONS

Some selected presentations on Rocks and selected projects. April 2011 Rocks Virtual Clusters, Extended Clusters into Amazon EC2 with Condor More »

Service Pack Roll for Rocks 6.1 Released

MARCH 11, 2013 ADMIN

The Service Pack Roll for Rocks 6.1 is now released.

This fixes a number bugs/errors/omissions in Rocks 6.1 (Emerald Boa). All new clusters should be built

READERS COMMENTS

Recent Posts | Comments | Pages

Service Pack Roll for Rocks 6.1 Released
March 11, 2013 By admin

Rocks 6.1 (Emerald Boa) for i386 and x86_64 Released!
November 29, 2012 By admin

Rocks 6.1 (x86-64) Beta (Emerald Boa) Available
October 23, 2012 By admin

Rocks Mamba (5.5/6.0) is Released
May 09, 2012 By admin

Rocks Version 5.5/6.0 (Mamba) Beta Slated for End of March 2012
March 17, 2012 By admin

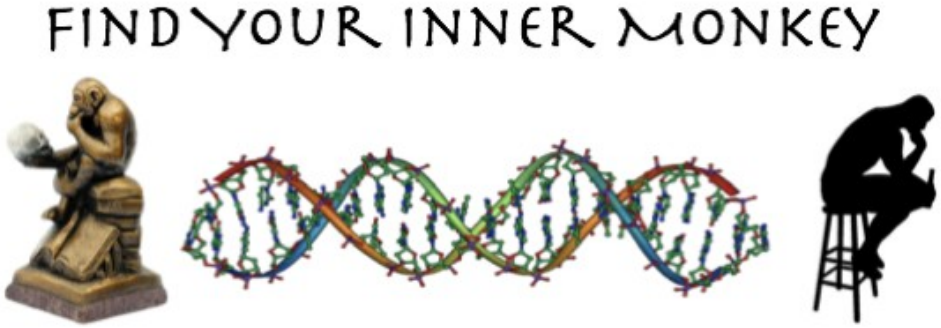
MONTHLY ARCHIVE

Select Month

http://www.rocksclusters.org

Rocks

Home | Downloads | Support and Docs | Rocks-A-Palooza | Register Your Cluster | Licensing | About



February 1, 2011

NBCR Releases APBS Roll for Rocks 5.4

The National Biomedical Computation Resource at University of California, San Diego is pleased to announce the availability of APBS Roll package for Rocks clusters version 5.4 for the x86_64 architecture.

APBS (Adaptive Poisson-Boltzmann Solver) is a scalable Poisson-Boltzmann equation solver used to study electrostatic properties of small to nanoscale biomolecular systems. The APBS Roll simplifies APBS deployment and integration on Rocks clusters. More information about APBS can be found at SourceForge or at the NBCR web site.

This APBS Roll contains the latest APBS version 1.3 and PDB2PQR package version 1.7.

The APBS Roll can be downloaded from the APBS download site and the Roll documentation including installation and usage information is available here.

Questions or comments may be directed to Robert Konecny <

Recent News

- Service Pack Roll v5.4.2 for Rocks v5.4 is Released
- PVFS2 Roll for Rocks v5.4 is Released
- Service Pack Roll v5.4.1 for Rocks v5.4 is Released
- University of Tromso Releases the Torque Roll for Rocks 5.4
- Rocks v5.4 (Maverick) is released for Linux on the i386 and x86_64 CPU architectures
- Rocks v5.4 Beta is released for x86_64
- Rocks-A-Palooza V Registration is Open
- University of Tromso Releases the Torque Roll for Rocks 5.3

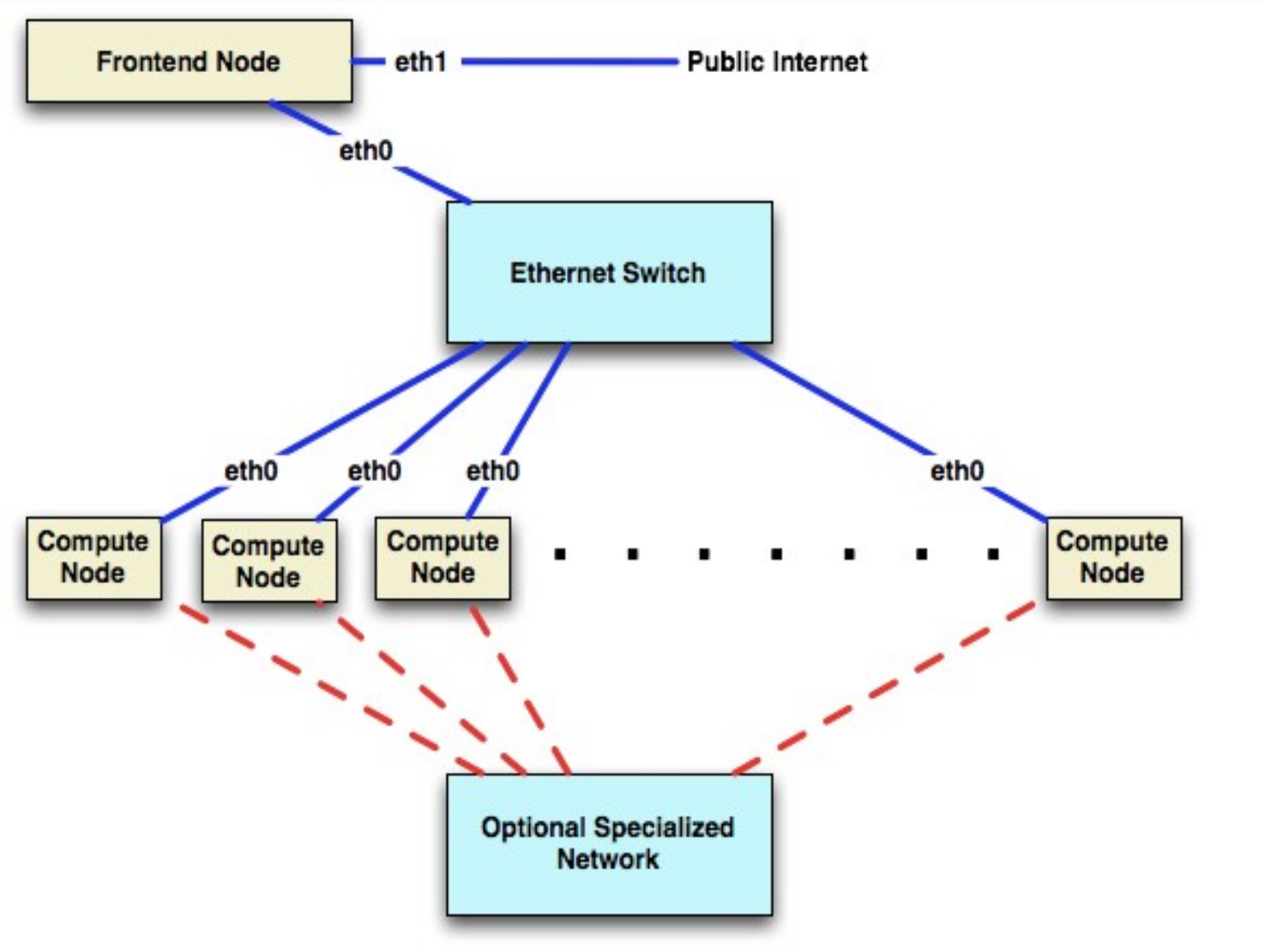
Archives

- February 2011
- January 2011
- December 2010
- November 2010

ROCKS

- NPACI Rocks is a collection of open source software for building a HP Cluster released by the San Diego Supercomputer Center at the University of California, San Diego
- The primary design goal for Rocks is to make cluster installation as easy as possible
- When you install Rocks, you'll install both the clustering software and a current version of RedHat Linux updated to include security patches.
- The Rocks installation will correctly configure various services.
- Default installation tend to go very quickly and very smoothly.

ROCKS Cluster scheme



Rocks

- A Rocks cluster has the same basic architecture as an OSCAR cluster. The head node, or frontend, is a server with two network interfaces.
- You'll install the frontend first and then use it to install the compute nodes. The compute nodes use HTTP to pull the RedHat and cluster packages from the frontend
- The ISO images can be downloaded from http://www.rocksclusters.org/wordpress/?page_id=3
- Athlon, Pentium, i386, Xeon and Opteron architectures are supported

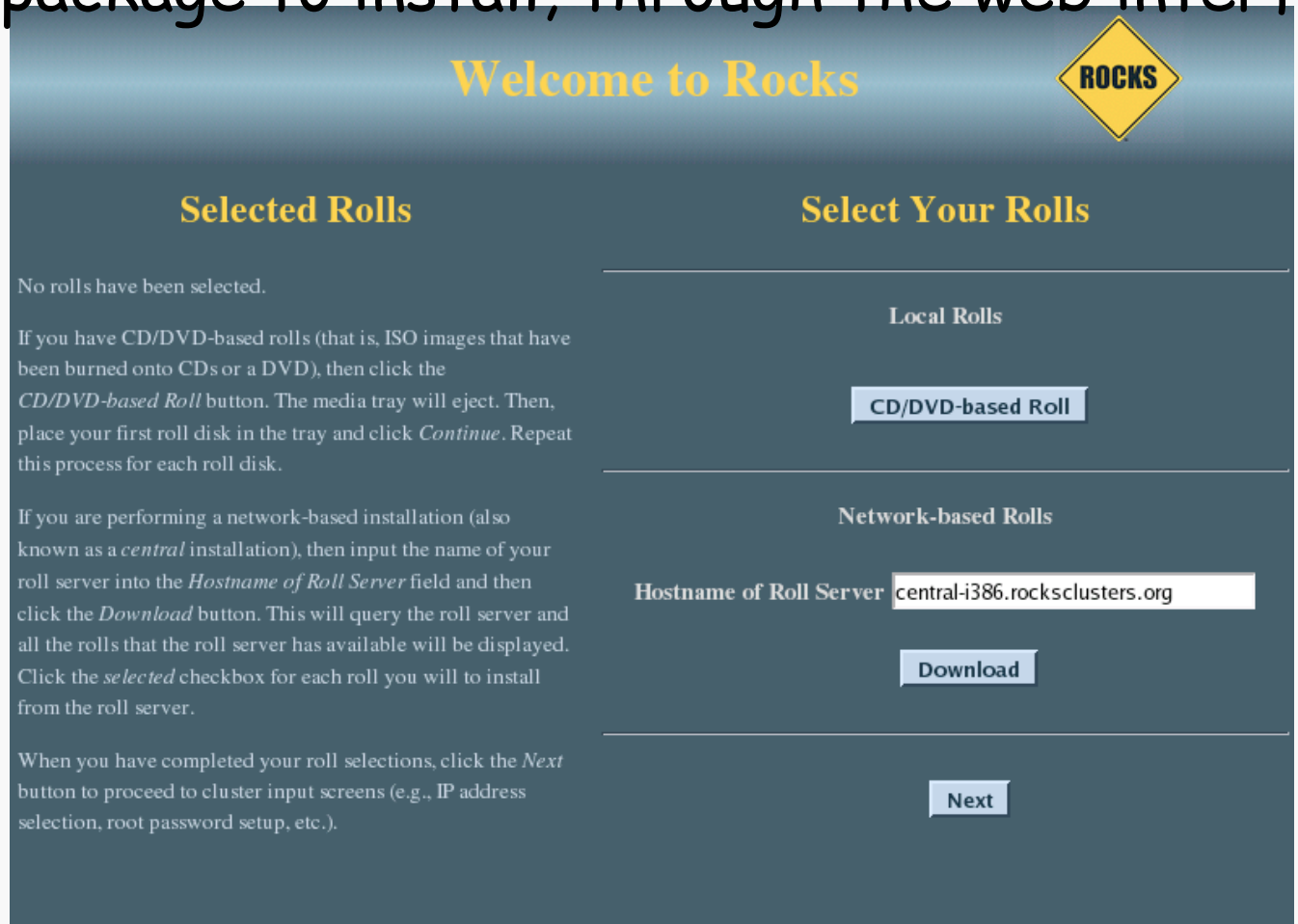
Rocks

Initial image after inserting the CD. To start the installation, type:

```
build
```

Rocks installation

- After some basic textual screens to enter the network config, the user is request to specify the various package to install, through the web interface.



Rocks tools -
Active node configuration management

- Nodes are installed using the RedHat kickstart tool which is driven by a text-based configuration file. This file contains all the package names to install as well as post-processing commands. In Rocks the kickstart files are dynamic, i.e. they are actively managed by building them on-the-fly with a CGI script. The script's functions are:
  - to construct a general configuration file from a set of XML-based configuration files; and
  - to apply node-specific parameters by querying a local SQL database

Rocks tools -
Active node configuration management

- There are two types of XML-based configuration files:
  - nodes - small, single-purpose modules that specify packages and per-package post-configuration commands for a specific service;
  - graphs - files that link the defined modules together using directed edges, where an edge represents a relation between two modules.
- The roots of the graph represent "appliances" such as `compute` and `frontend`. This XML-based installation infrastructure describes all node behaviors.

Rocks tools -
Active node configuration management

- The installation procedure involves the following steps:
  - a machine requests its kickstart file via HTTP from a CGI script on the frontend server;
  - the script uses the requesting node's IP address to drive a series of SQL queries that determine the appliance type, software distribution, and localization of the node;
  - the script parses the XML graph and traverses it, parsing all the node files based on the appliance type;
  - a RedHat compliant text-based kickstart file is returned to the requesting machine.
- This method is very flexible, allowing heterogeneous hardware to be supported as easily as homogeneous hardware.

Rocks

- Every installation will require the Rocks Base and the HPC optional packages. The core install provides several flavours of MPICH, Ganglia, and PVFS.
- Additional software (called rolls) can be installed from the optional CDs (not all rolls are available for all architectures)
  - Sun Grid Engine (SGE) Roll
  - Grid roll (NSF Middleware Initiative, NMI)
  - Intel roll (Intel compilers)
  - Area 51 roll (`tripwire` and `chkroot.tripware`)
  - Scalable Cluster Environment (SCE) roll
  - Java roll
  - PBS roll
  - Condor roll

Installing the frontend

- Boot the frontend with the Rocks Base CD and stay with the machine. After a moment, you'll see a boot screen giving you several options. Type `frontend` at the `boot:` prompt and press Enter.
- You need to do this quickly because the system will default to a compute node installation after a few seconds and the prompt will disappear.
- After a brief pause, the system asks you to register your roll CDs, asking you wheter you have any roll CDs: click yes . Repeat for all roll CDs you have. Registration is now done, but at the end of the installation you'll be prompted for these disks again for the purpose of actual software installation

Installing the frontend

- The next screen prompts you for information that will be included in the web reports that Ganglia creates. This includes the cluster name, the cluster owner, a contact, a URL, and a latitude and longitude for the cluster location. These information are not visible over the public interface.
- Next the partitioning of the disk drive is performed
- The next few screens are used to configure the network interfaces. The last network setup screen asks for a host name. The frontend name will be written to a number of files, so it is very difficult to change it,

Installing the frontend

Virtual consoles

Console	Use	Keystroke
1	Installation	Cntl-Alt-F1
2	Shell prompt	Cntl-Alt-F2
3	Installation log	Cntl-Alt-F3
4	System messages	Cntl-Alt-F4
5	Other messages	Cntl-Alt-F5

Installing the frontend

- You'l be prompted next for a root password.
- Then Rocks will format the filesystem and begin installing the packages. A status report will show the progress of the installation for each package.
- Ath the end of the packages installation, the frontend will reboot.
- The frontend is now installed. You can move onto the compute nodes

Install Compute Nodes

- Before performing the installation you may make a few changes to the defaults.
- You might want to change how the disks will be partitioned, what packages will be installed or which kernel will be used.
- To install the compute nodes, you'll begin by running the program `insert-ethers` as root on the frontend.
- Next, you'll boot a compute node using the Rocks Base CD.
- `insert-ethers` listen for a DHCP query from the booting computing node, assigns it the name, IP address end begin the installation of the client

Managing Rocks

- One of the Rocks' strengths is the web-based management tools it provides.
- Initially the HTTP traffic is blocked by the firewall. To allow access over the public interface, edit the `/etc/sysconfig/iptables` and uncomment the line:

```
-A INPUT -i eth1 -p tcp -m tcp –dport www -j ACCEPT
```

- Then restart iptables service

```
service iptable restart
```

- As an alternative you can access it from the frontend Xwindow at the URL `http://localhost`

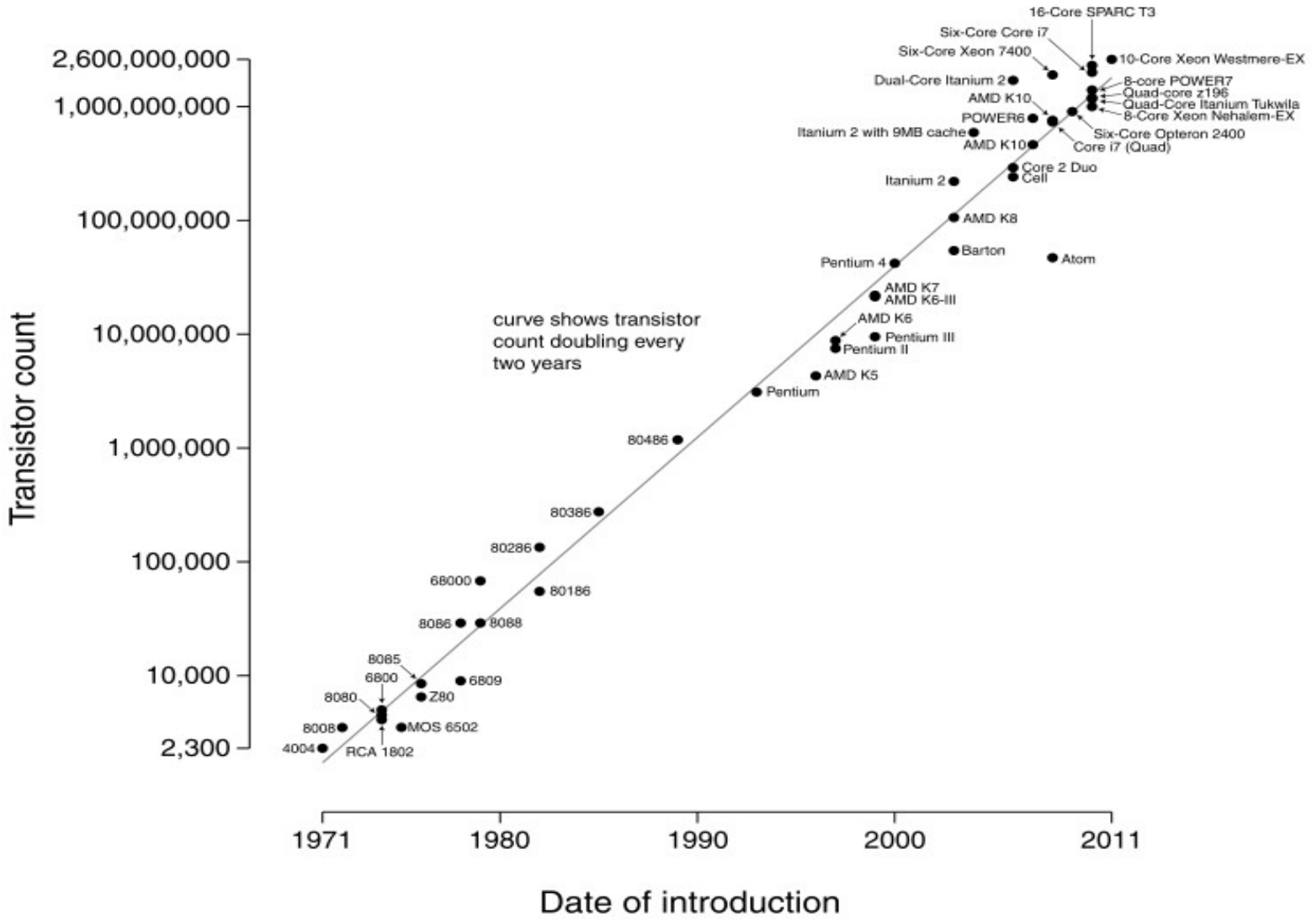
Rocks web interface

- The links on the web page will vary depending on the software or rolls you choose to install. The main options are:
  - Cluster database (SSL) – phpMyAdmin -
  - Cluster Status – Ganglia -
  - Cluster Top – Process viewer -
  - PBS Job queue
  - News – RSS -
  - Proc filesystem
  - Cluster distribution
  - Kickstart Graph
  - Roll Call
  - Rocks User's Guide / Reference Guide

New frontieres

- Arndale Exynos board
- Mont-Blanc EU project
- Rasberry Pi

Moore's law

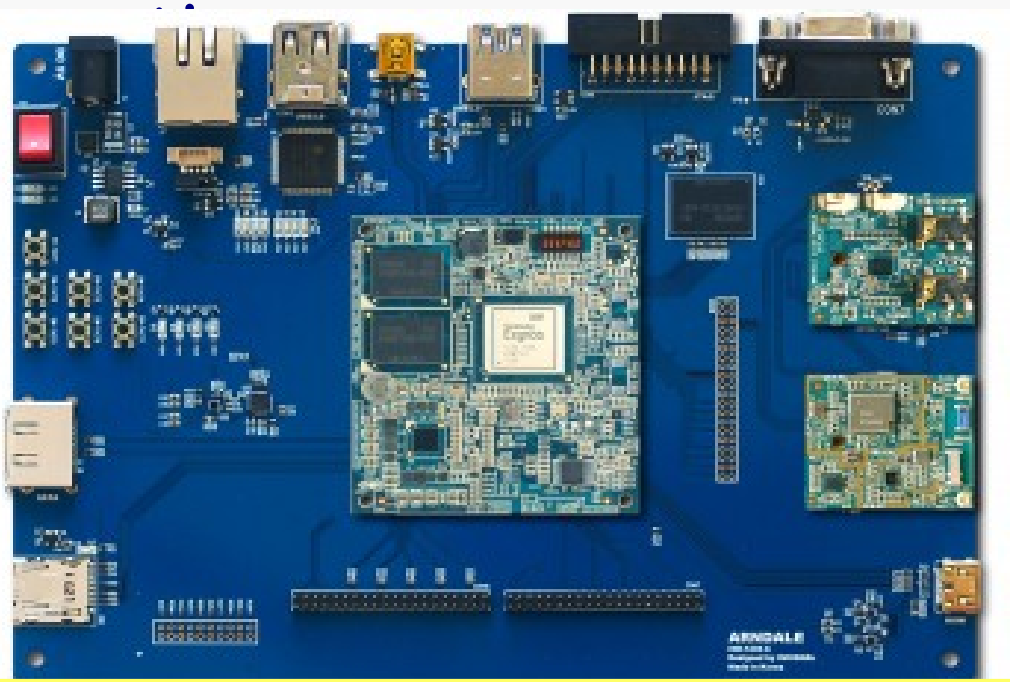

Microprocessor Transistor Counts 1971-2011 & Moore's Law
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
curve shows transistor count doubling every two years
16-Core SPARC T3
Six-Core Core i7
Six-Core Xeon 7400
10-Core Xeon Westmere-EX
Dual-Core Itanium 2
8-core POWER7
AMD K10
Quad-core z196
Quad-Core Itanium Tukwila
POWER6
8-Core Xeon Nehalem-EX
Itanium 2 with 9MB cache
Six-Core Opteron 2400
AMD K10
Core i7 (Quad)
Core 2 Duo
Cell
Itanium 2
AMD K8
Barton
Atom
Pentium 4
AMD K7
AMD K6-III
AMD K6
Pentium III
Pentium II
AMD K5
Pentium
80486
80386
80286
68000
80186
8086
8088
8085
6800
6809
8080
Z80
8008
MOS 6502
4004
RCA 1802
1971
1980
1990
2000
2011
Date of introduction

Heterogeneous Computing on Arndale board

- Heterogeneous Computing is the transparent use of all computational devices to carry out general purpose scientific and engineering c

Android and Ubuntu support

Enables Google Open Source project



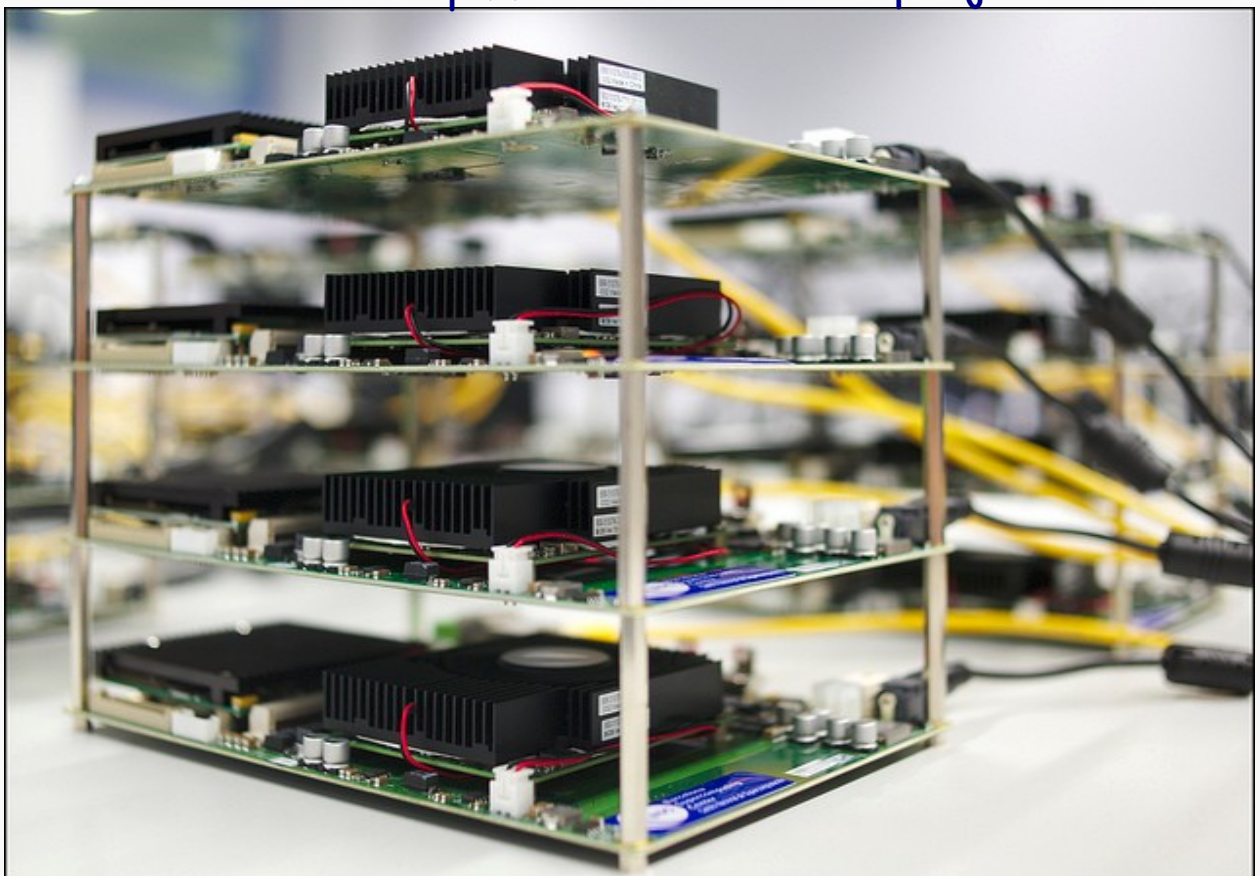
The **Arndale** Board based on **ARM Cortex-A15** with **Mali-T604** **Samsung Exynos 5250** development platform

Very promising architecture for hererogeneous computing: it is built on **32nm low-power HKMG (High-K Metal Gate)**, and features a dual-core **1.7GHz mobile CPU built on ARM® Cortex™-A15** architecture plus an integrated **ARM Mali™-T604 GPU** for increased performance density and energy efficiency

The future of Super Computing Centers: the MontBlanc EU project

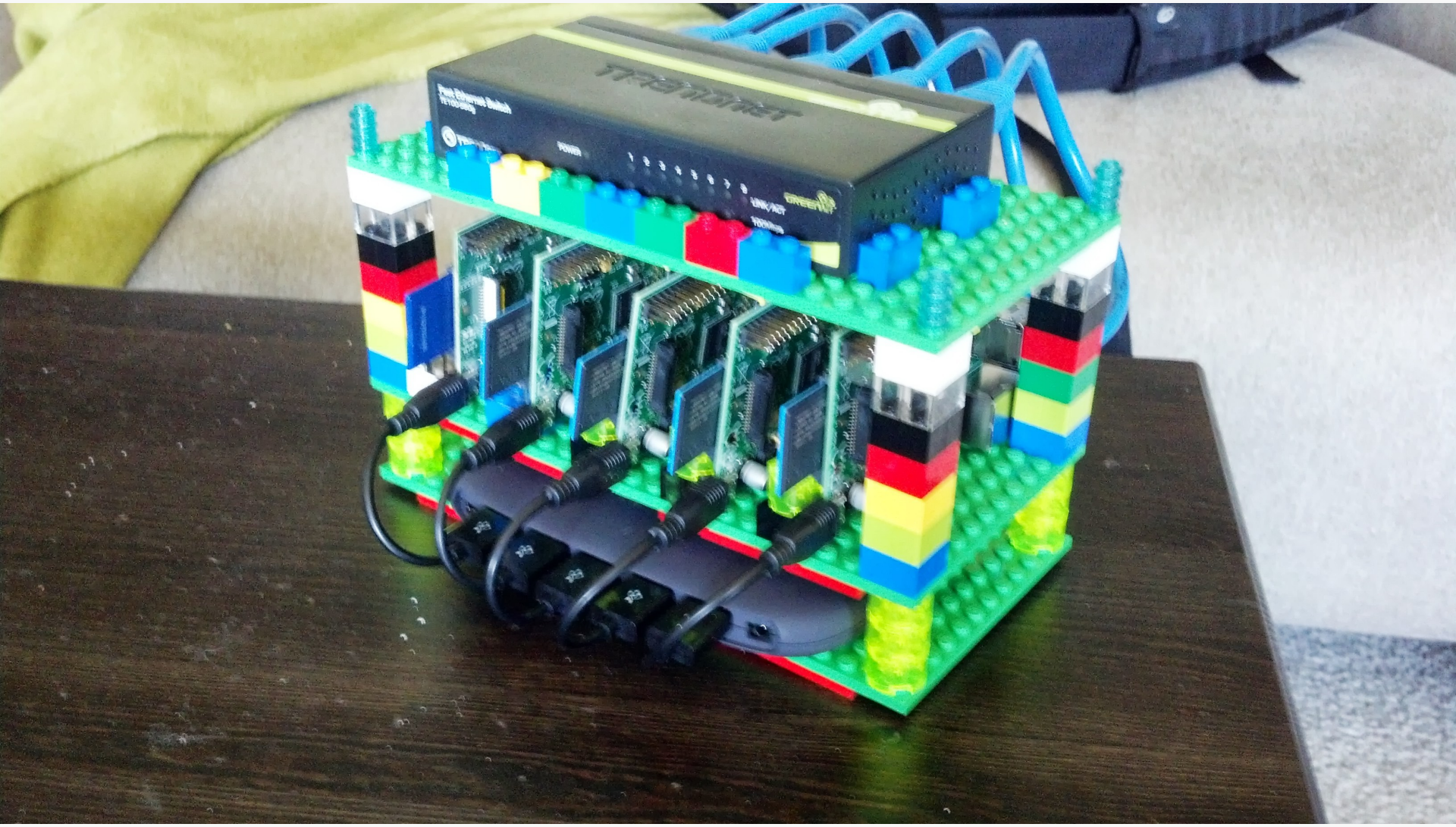
- Heterogeneous Computing and minimization of power consumption: the new HPC Center of the future!

http://www.montblanc-project.eu



MontBlanc selected the Samsung Exynos 5 Processors

Building Raspberry Pi clusters



Source: Adam DeConinck http://blog.ajdecon.org/building-tiny-compute-clusters-for-fun-and-learning/

Raspberry Pi

- Credit Card sized computer that plugs into your TV and a keyboard.
- It is a capable little computer which can be used in **electronics projects**, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays **high-definition video**. We want to see it being used by kids all over the world to learn programming.

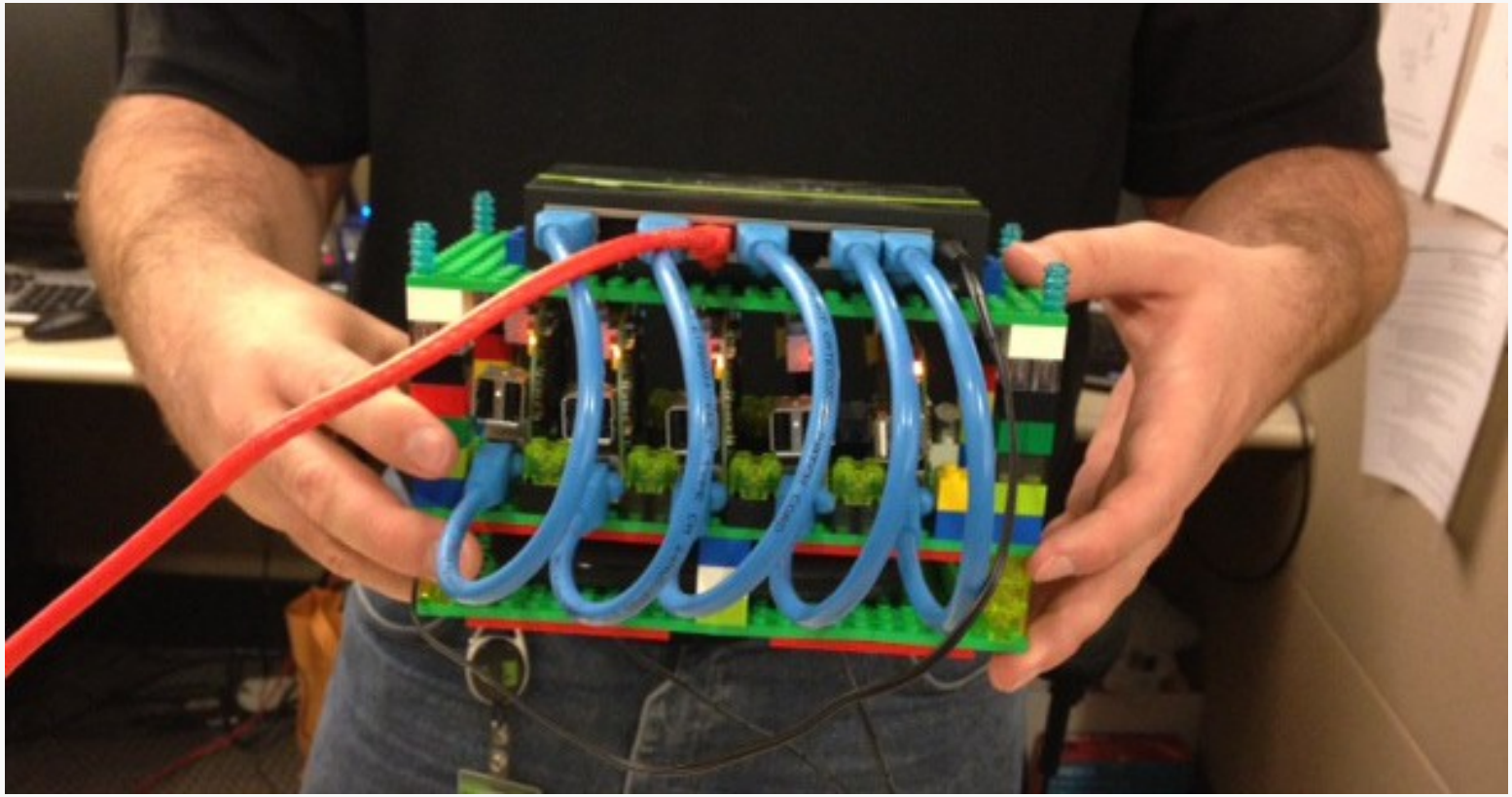
Costs:
Model A: 25$
Model B: 35$

Plus taxes, shipping, etc..

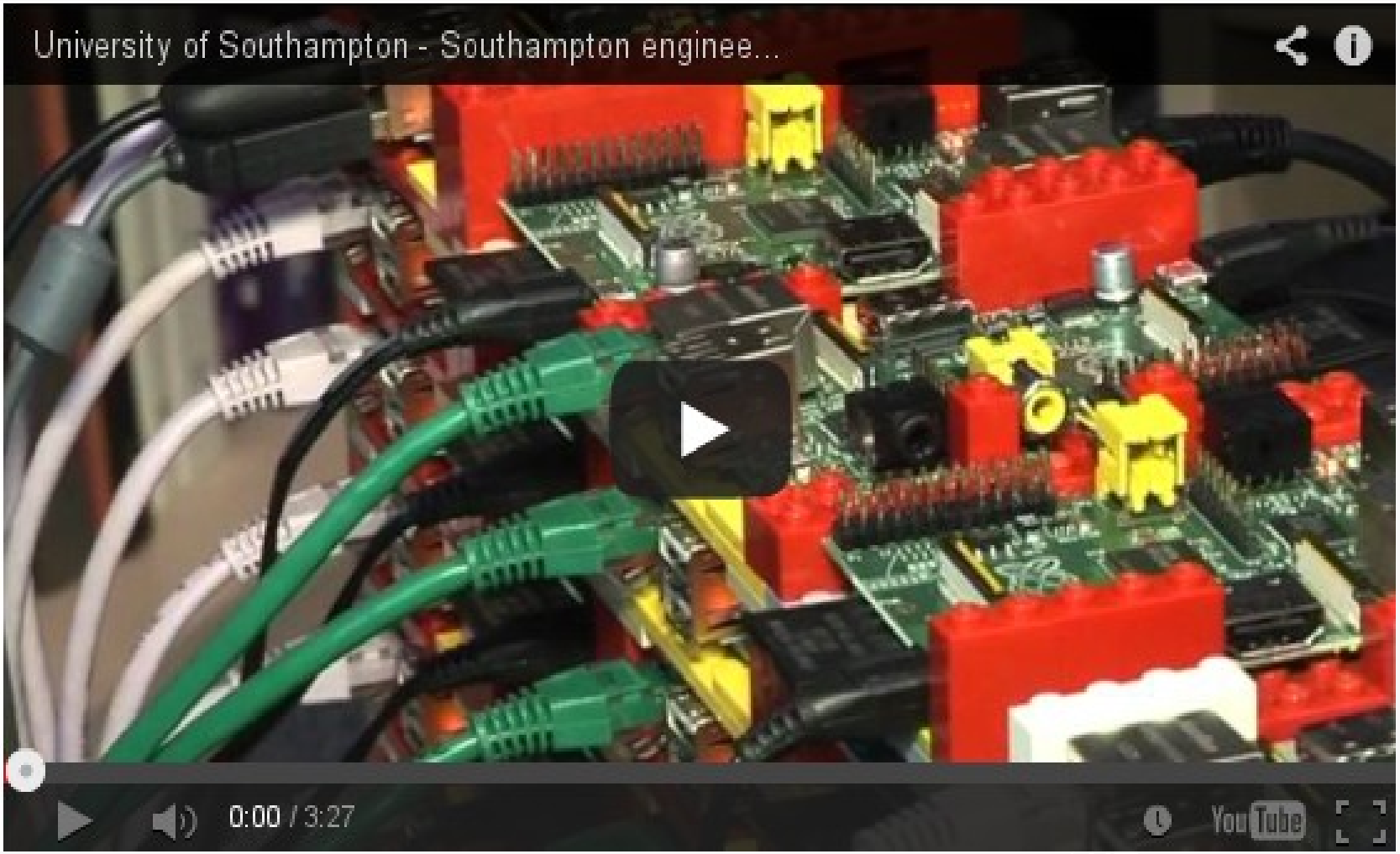
Power supply and the SD card are not included

Assembled using Lego

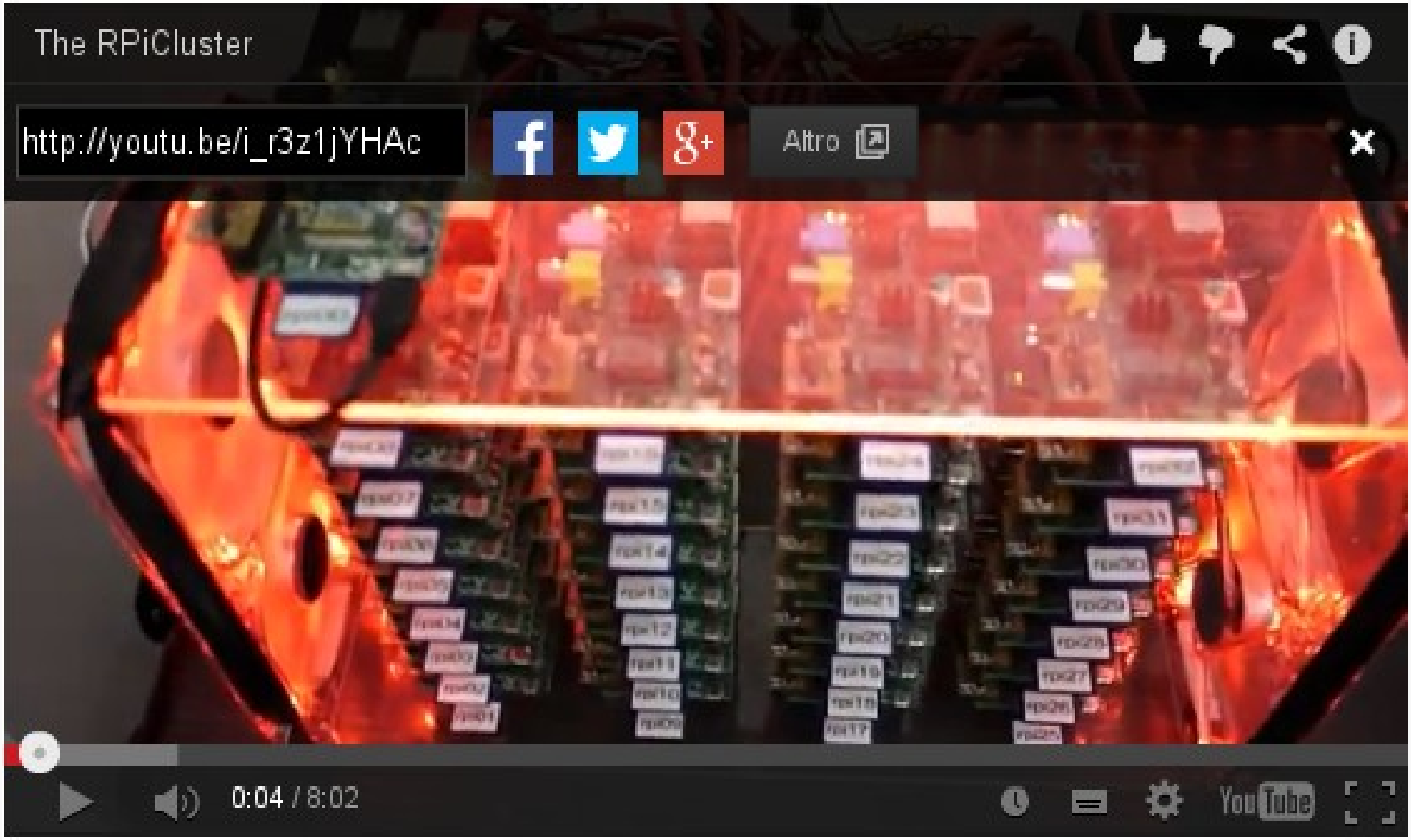
Raspbian Linux, SLURM scheduler, Ganglia Monitor, SSH, Ansible (Playbooks), OpenMPI
670MFLOPS!!!



Idris-Pi
Southampton University



RPiCluster



SantaFe

Bobo

- Bobo is the largest Raspberry Pi cluster
- Bobo costs 3800€
- Bobino costs 900€

Prof. Pekka Abrahamsson
Libera Università Bolzano

LittleFe